GESTIÓN DE REDES DE DATOS A TRAVÉS DE ONTOLOGÍAS UTILIZANDO SISTEMAS MULTIAGENTES

WILLIAM HERNANDEZ CHICA

Director

NESTOR DARÍO DUQUE MENDEZ

UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MANIZALES
FACULTAD DE INGENIERIA Y ARQUITECTURA
MAESTRIA EN AUTOMATIZACIÓN INDUSTRIAL
2015

MANAGEMENT DATA NETWORKS THROUGH ONTOLOGIES USING MULTIAGENT SYSTEMS

WILLIAM HERNANDEZ CHICA

Director

NESTOR DARÍO DUQUE MENDEZ

UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MANIZALES
FACULTAD DE INGENIERIA Y ARQUITECTURA
MAESTRIA EN AUTOMATIZACIÓN INDUSTRIAL
2015

RESUMEN

En una red de transmisión de datos, la utilización de ontologías como técnica para la representación del conocimiento entrega la capacidad de compartir conceptos afines entre sus equipos, aplicaciones y servidores. La aplicación de conceptos en el campo aplicados en el campo de la gestión de red genera ventajas que permite la aplicación a la integración de múltiples modelos de información de gestión de red, especialmente en entornos tan heterogéneos como éste. Así, existirá una correspondencia entre las definiciones de los diferentes modelos de información, facilitando la ardua tarea de gestionar la red.

Un Sistema Multiagente es la pieza clave para la adecuada gestión de red. La aplicación de una ontología multimodal permite a este sistema facilitar la labor del administrador de red, mediante la integración de múltiples modelos de información. Para llevar a cabo la integración, las ontologías son interoperables entre los elementos de gestión de red, utilizando diferentes niveles de abstracción. De esta forma, el administrador realizará exclusivamente las tareas correspondientes a la gestión de la infraestructura.

Esta tesis tiene como objetivo representar mediante ontologías los diferentes tipos de equipos, servidores, dispositivos que hacen parte de una red de computadores, y adicionalmente definir los conceptos para que los administradores de red puedan realizar una gestión a través de un sistema basado en ontologías, de esta forma lograr la interoperabilidad entre los diferentes dominios y elementos de gestión.

Palabras Claves: Dominio, Ontología, Modelo, Interoperabilidad, Semántica, Comportamientos, Integración, OWL, Agentes Inteligentes.

ABSTRACT

In a network of data transmission, the use of ontologies as a technique for representing knowledge delivers the ability to share related concepts among its hardware, software and servers. Applying concepts applied in the field in the field of network management generates advantages allowing application to the integration of multiple models of network management information, particularly in heterogeneous environments such as this. So, there will be a correspondence between the definitions of the different models of information, facilitating the arduous task of managing the network.

A Multi-Agent System is the key to proper management network. The application of a multimodal ontology allows the system administrator to facilitate the work of the network by integrating multiple information models.

To carry out the integration, ontologies are interoperable between elements of network management, using different levels of abstraction. Thus, the administrator perform the tasks exclusively to the management of the infrastructure.

This thesis aims ontologies represented by different types of computers, servers, devices that are part of a computer network, and further define the concepts so that network administrators can perform management through a system based on ontologies, thus achieving interoperability between different domains and management elements.

Keywords: Domain , Ontology , Model, Interoperability, Semantics, Behaviors, Integration, OWL, Intelligent Agents .

CONTENIDO

INTRODUCCIÓN

Desde la década de los 80s las redes de datos han evolucionado para suplir las necesidades de un mundo que necesita cada día más conexiones, lo cual ha hecho que se conviertan cada día en sistemas más complejos. Esta tendencia al incremento en complejidad y tamaño de las redes obtuvo como consecuencia un aumento de los costes de administración y manejo, debido a que era necesario mayor cantidad de personal cualificado para su gestión y control eficiente de sus recursos. Para atender estos inconvenientes se tuvo la necesidad de disponer de herramientas que facilitaran las labores de administración de red, entre otros, como: capacidad de gestión de elementos, mayor disponibilidad, resolución y supervisión de problemas.

Al comienzo, los productores de equipos de telecomunicaciones diseñaron soluciones de gestión que sólo funcionaban sobre sus redes propietarias, de forma que el administrador de red, con equipos y elementos de varios fabricantes se hallaba con un problema de diversidad tecnológica, con diferentes y heterogéneos dominios de gestión.

Con la necesidad de minimizar la dependencia de la gestión con respecto al fabricante, se desarrollaron los modelos de Sistemas Unificados de Gestión de Redes, que proponen una serie de elementos estandarizados y abiertos, con el objetivo de que éstos fuesen adoptados o asimilados por los diversos fabricantes, facilitando de esta forma la interconexión y gestión de un equipo de cualquier fabricante. Pero este desarrollo no trajo consigo la unificación total de todos los sistemas en un solo paradigma de Gestión, por el contrario los agrupó en segmentos, de esta forma OSI se utiliza sobre todo para la gestión de equipos y servicios en redes de telecomunicación a través de la arquitectura TMN (Telecommunication Management Network, Red de Gestión de Telecomunicaciones), SNMP se utiliza ampliamente en Internet y está más orientado a la gestión de equipos y servicios de redes de datos. En la Tabla 1 se presentan algunos de los modelos más relevantes.

Tabla 1. Modelos Relevantes

ENTORNO	PROTOCOLO	LENGUAJE	MODELOS DE INFORMACIÓN
TMN/OSI	CMIP	GDMO	X.721, M.3100
Internet	SNMP	SMI	MIB-II, otros
Desktops	DMI	MIF	Master MIF

Distributed Processing	CORBA/ IIOP	IDL	X.780, M.3120
Web Based Management	HTTP/XML	MOF/CIM	CIM Schemas

TMN está enfocado a sistemas de comunicaciones en general, como telefonía….
Y SNMP está enfocado a los servicios de equipos de red Ethernet, el cual provee los servicios de conectividad via internet.

En los siguientes capítulos se presentan los modelos de gestión, cada uno de ellos propone su propio mecanismo de intercambio, su lenguaje de definición y su conjunto de definiciones de información. Aunque cada uno es capaz de gestionar dispositivos de diferentes fabricantes, son modelos incompatibles entre sí, y su heterogeneidad impide alcanzar la meta de la gestión integrada, que es conseguir gestionar los distintos recursos desde una misma herramienta de gestión.

PLANTEAMIENTO DEL PROBLEMA

A través del tiempo han aparecido diversos modelos de Sistemas Unificados de gestión de redes especificados por diferentes organismos de estandarización. Cada uno de ellos se ha emitido asociado a un ámbito o contexto determinado, pero con la evolución de las redes y su progresiva convergencia de servicios y aplicaciones, los ha hecho diverger en diferentes sistemas de gestión. Cada uno de ellos propone su propio mecanismo de intercambio, su lenguaje de definición y su conjunto de definiciones de información. Por esta razón, cada uno es incompatible con el resto, y su heterogeneidad impide alcanzar la meta de la gestión integrada, que es conseguir gestionar los distintos recursos desde una misma herramienta de gestión.

Para allanar estos inconvenientes se plantea un modelo de una arquitectura de un gestor de red basado en ontologías que permita la interoperabilidad entre los modelos de gestión definidos por los diferentes organizamos de gestión y el nuevo modelo común aprovechando las posibilidades que ofrecen la utilización de ontologías para la integración semántica de dichos modelos.

1 MODELOS DE GESTIÓN

La gestión de sistemas interconectados se compone de todas las medidas necesarias para asegurar la operación efectiva y eficiente de un sistema y sus recursos según los objetivos de una organización.

La Gestión de Red es Integrada surge como consecuencia de la necesidad de gestionar equipos de telecomunicación heterogéneos, y consiste en la posibilidad de gestionar diferentes tipos de recursos de red (recursos de transmisión y conmutación) procedentes de diversos fabricantes utilizando un mismo conjunto de herramientas de gestión. Para ello, la gestión integrada usa conceptos estandarizados de bases de datos de gestión globales; permite una aproximación integral a distintos aspectos, incluyendo los organizacionales, soporte de sistemas heterogéneos; y ofrece programación abierta e interfaces de usuario.

Así con esta definición se presentan los modelos de gestión que presentan las diferentes entidades de estandarización y los veremos en las siguientes secciones.

1.1 MODELO DE GESTIÓN OSI Y ARQUITECTURA TMN

La Organización Internacional de la Estandarización, ISO (International Standarization Organization,) definió el modelo OSI-SM (Open Systems Interconnection — Systems Management, Interconexión de Sistemas Abiertos — Gestión de Sistemas) con el objetivo de proporcionar una estructura de red organizada para conseguir la interconexión de los diversos tipos de Sistemas de Operación y equipos de telecomunicación usando una arquitectura estándar e interfaces normalizados. [1]

El modelo OSI-SM posee una serie de características realmente útiles, de forma que la ITU (International Telecommunication Union, Unión Internacional de Telecomunicaciones) decidió utilizarlo como base para su propuesta de arquitectura de gestión TMN (Telecommunication Management Network, Red de Gestión de las Telecomunicaciones). Este modelo TMN define una arquitectura física (estructura y entidades de la red), un modelo organizativo (niveles de gestión), un modelo funcional (servicios, componentes y funciones de gestión) y un modelo de información (definición de recursos gestionados), alcanzando una considerable difusión entre los operadores de tecnologías de telecomunicaciones.

TMN reutiliza muchos conceptos del modelo de gestión propuesto por ISO2 (International Organization for Standardization, Organización Internacional para la Estandarización) en OSI-SM (Open Systems Interconnection - Systems

Management, Interconexión de Sistemas Abiertos - Gestión de Sistemas) También define los siguientes conceptos de gestión:

- Es un mecanismo de comunicaciones basado en CMIS (Common Management Information Service, Servicio Común de Información de Gestión) además de CMIP (Common Management Information Protocol, Protocolo Común de Información de Gestión).

- El lenguaje GDMO (Guidelines for the Definition of Managed Objects, Directrices para la Definición de Objetos Gestionados), mediante el cual se especifica la información de gestión del Modelo de Información Genérico contenido en la recomendación M.3100.

1.1.1 Protocolo de Comunicaciones: CMIS Y CMIP

CMIP [2] define la arquitectura de comunicación del modelo, el protocolo de comunicaciones. Tanto el gestor como el agente se sitúan en nivel de aplicación OSI, incluyendo una serie de bibliotecas de funciones. CMIP es un protocolo orientado a conexión, aportando una mayor fiabilidad pero introduciendo a la vez una sobrecarga en las comunicaciones, lo que aporta confiabilidad con detrimento de la simplicidad y generando mayor retardo en las operaciones.

CMIS [3] define los servicios de comunicaciones disponibles o primitivas. Existen dos tipos:

- Servicios de operación: Son servicios usados por el gestor para invocar operaciones de gestión a los agentes y permitir a éstos devolver los resultados de dichas operaciones a los gestores. Dichos servicios son M-GET (pedir información de los objetos gestionados), M-SET (modificar o añadir información en los objetos), M-ACTION (llevar a cabo una acción por parte de un objeto), M-CREATE (crear un nuevo objeto gestionado), M-DELETE (eliminar un objeto gestionado) y M¬CANCEL-GET (cancelar una petición anterior). Asimismo, se han definido una serie de métodos para poder invocar dichos servicios sobre un conjunto de objetos distintos: alcance (especifica el conjunto de objetos sobre los que aplicar la operación), filtrado (restringe la operación según el valor de una lista de atributos) y sincronización (especifica que la operación sea de tipo atómico o bien best effort).

- Servicios de notificación Se usa cuando un agente quiere informar a un gestor acerca de una notificación generada por un determinado objeto gestionado. Hay que informar del modo de conexión, del objeto que la ha generado, del tipo de notificación y otros parámetros con información adicional. Para este servicio de

notificaciones va a existir una clase de objetos gestionados denominada discriminador de eventos (EFD, Event Forwarding Discriminator), que implementa un mecanismo de filtrado de las notificaciones que recibe y posee un atributo que contiene la dirección del agente destino. También suele existir otro tipo de objetos que registran las notificaciones y poseen un mecanismo similar de filtrado, pero en este caso, para almacenar o no las notificaciones. Dado que la gestión OSI distribuye la carga de la gestión entre gestor y agentes permitiendo a los últimos enviar notificaciones al primero, este servicio es uno de los que más caracteriza a este modelo.

Ambos servicios funcionan de forma cliente-servidor, pero con la diferencia de que para los servicios de operación el servidor es el agente y en la parte de notificación el gestor.

1.2 MODELO DE GESTIÓN DE RED SNMP

El modelo de gestión de red integrada definido por el IETF3 (Internet Engineering Task Force, Grupo de Trabajo de Ingeniería para Internet) para Internet es el conocido por el nombre de su protocolo, SNMP (Simple Network Management Protocol, Protocolo Simple de Gestión de Red). Este modelo también define el lenguaje de definición de información SMI (Structure of Management Information, Estructura de la Información de Gestión). Además, existe un gran conjunto de MIBs estándar que se han ido definiendo a lo largo de los años de existencia de SNMP. Se basa en los siguientes elementos [4]:

- El protocolo SNMP (Simple Network Management Protocol, Protocolo Simple de Gestión de Red), que define las comunicaciones entre gestor y agente.

- El lenguaje de definición de información de gestión SMI (Structure of Management Information, Estructura de la Información de Gestión), que normaliza la sintaxis.

- Un modelo de información basado en MIBs (Management Information Bases, Bases de Información de Gestión). El IETF ha definido numerosas MIBs estándar con el objeto de proporcionar una serie de conceptos comunes, y sigue definiendo más, avanzando hacia la normalización semántica.

Dado que la filosofía del modelo OSI es dotar a los agentes de un potente mecanismo de notificaciones, que permite balancear la carga de procesamiento entre gestor y agente, ahora la del modelo de gestión SMNP es justamente la contraria: la capacidad de gestión de red a los nodos debe ser mínimo. Este

planteamiento lo hace coherente con el modelo TCP/IP, cuya característica principal es su simplicidad, y pueda implantarse en todo tipo de recursos gestionados.

El protocolo SNMP es el heredero de SGMP (Simple Gateway Monitoring Protocol, Protocolo Simple de Supervisión de Pasarelas), definido a finales de los ochenta como propuesta para realizar tareas de gestión en Internet. Desde su aparición se han desarrollado diversas versiones, que buscaban intentar mejorar la funcionalidad y el modelo administrativo, pero no siempre han tenido una implementación deseada, lo cual hace que coexistan en la actualidad la versión 1, la versión 2C, que incluye mejoras en la definición rendimiento del protocolo y de la información, pero sigue basado su modelo administrativo en el manejo a través comunidades, y la versión 3, que añade funciones relativas a la seguridad. Manteniendo en todas las versiones una filosofía de la simplicidad.

Las operaciones que permiten efectuar operaciones están restringidas por simplicidad a la modificación, obtención y notificación de información. Un gestor SNMP puede enviar a un agente tres tipos de peticiones: GetRequest, GetNextRequest y SetRequest. A partir de la versión 2, se incluye además GetBulkRequest, que mejora el rendimiento al solicitar información. Todas son confirmadas por el agente mediante un mensaje del tipo Response. Además, los agentes pueden enviar en situaciones críticas un mensaje no solicitado o Trap que no será confirmado. También pueden enviarse mensajes informativos entre gestores, similares a las Traps pero que sí son confirmados. Hay que destacar que las notificaciones SNMP no tienen la misma funcionalidad que los eventos de la gestión OSI. En SNMP su propósito es el de informar de situaciones muy específicas y no están pensados para llevar a cabo una gestión orientada a eventos. Utilizando inteligentemente el modelo de información se pueden añadir otras operaciones. Por ejemplo, a través de la definición de una columna de tipo RowStatus se pueden crear y borrar filas en las tablas [4].

En lo que se refiere a la arquitectura de protocolos, el protocolo SNMP se basó inicialmente en UDP, reservándose los puertos 161 y 162 para acceder a los agentes y gestores respectivamente. Debido a su éxito se ha especificado su transporte en protocolos de redes OSI, Apple y Novell. Para la codificación con sintaxis neutra se utiliza un subconjunto de ASN.1, y a partir de la versión 3, se emplea MD.5 (Message Digest , Resumen de Mensaje 5) para autenticar y DES (Data Encryption Standard, Estándar de Cifrado de Datos) en su modo CBC (Cipher Block Chain, Encadenado de Bloques Cifrados) para cifrar los mensajes[4].

1.2.1 SMI: [5]

Es un lenguaje de especificación de información de gestión que pretende ser lo más simple posible, pero sin renunciar a la mayoría de funcionalidades importantes de gestión de red.

Una de las razones de esta simplicidad es que no usa un modelo como el de la programación orientada a objetos. Las definiciones contienen objetos, especificados con macros ASN.1, pero sus tipos sólo pueden ser grupos de variables escalares y celdas de tablas. A pesar de no ser un modelo orientado a objetos común, las tablas se pueden ver como clases, donde los atributos son las columnas de la tabla, y cada fila contiene a un ejemplar de la clase.

Las macros del lenguaje son las siguientes [5]:

- Módulo (MODULE-IDENTITY). Define un módulo de información de gestión o MIB. Dicho módulo poseerá propiedades tales como su nombre y OID, las fechas de revisión y última actualización, la organización y datos de contacto de quién lo definió y una descripción textual acerca de la información incluida en dicho módulo.
- Tipo de objeto (OBJECT-TYPE). Forman lo que serían atributos de una clase, e incluyen nombre, OID, acceso (ninguno, notificación, lectura, escritura, creación), estado (actual, obsoleto, caducado) y descripción textual. Los tipos de datos que definen la sintaxis del objeto son los tipos ASN.1 INTEGER (y sus derivados Counter32, Gauge32, Unsigned32, Counter64, TimeTicks), OBJECT IDENTIFIER OCTET STRING (y sus derivados, como IpAddress u Opaque) y BITS. Se puede incluir en algunos casos la longitud o los valores máximo y mínimo, así como cierta semántica (si el número entero es un contador o un medidor, o si es una enumeración de valores). Además, los tipos de datos se pueden redefinir con la macro TEXTUAL-CONVENTION, no incluida en el diagrama. Opcionalmente, también se pueden incluir unidades de medida, valor por defecto, referencia, e indicador de índice si el objeto define la entrada de una tabla.
- Identidad de objeto (OBJECT-IDENTITY). Representan OIDs con un significado específico. Para ello, se les dota de un nombre, un estado y una descripción, y opcionalmente, una referencia.
- Notificación (NOTIFICATION-TYPE). Las notificaciones se describen con un nombre, un OID, un estado, una descripción, y opcionalmente una referencia. También pueden tener una lista de los tipos de objetos que se enviarán en la notificación.
- Otras macros no incluidas en el diagrama son las declaraciones de conformidad que indican las opciones de implementación para distintos grupos de objetos y notificaciones.

1.2.2 MIBs [5]:

Las MIBs definidas en SMI y propuestas en RFCs (Request For Comments, Solicitud de Comentarios) del IETF suman más de doscientas. Además, hay que

contar las MIBs privadas propuestas por cada fabricante. El servidor SimpleWeb4 contiene todas las MIBs incluidas en RFCs, y punteros a gran parte de las privadas.

Las MIBs de SNMP se podrían clasificar según la información que definen en los siguientes grupos:

- Gestión de protocolos de comunicaciones [5]. Las MIBs que definen información relativa a protocolos de comunicaciones suman el mayor número, siendo la más importante la MIB-II y sus actualizaciones. En general miden datos estadísticos de los protocolos para hacer gestión de rendimiento.
- Gestión de sistemas [5]. Estas MIBs permiten gestionar cuestiones relativas a los sistemas y no a la comunicación. MIBs de este tipo son la HOST-RESOURCES MIB y la ENTITY-MIB. Muchas MIBs privadas también definen este tipo de recursos, ocurriendo que a veces la misma información puede encontrarse en varias MIBs diferentes, especificadas de forma distinta.
- Gestión distribuida [5]. Este grupo de MIBs permite delegar a un sistema remoto para que éste realice tareas de gestión. Incluye a RMON y sus extensiones, o las de DISMAN6 (Distributed Management Workgroup, Grupo de Gestión Distribuida del IETF).
- Gestión administrativa [5]. Son las MIBs que se han definido en los distintos modelos administrativos de SNMP que permiten configurar a las entidades gestionadas. Como ejemplo de este caso están las MIBs de SNMPv3.

En esta arquitectura los agentes son muy simples, sólo deben encargarse de informar sobre el status de los dispositivos y sus actualizaciones, mientras que la complejidad de la gestión y del procesamiento de los datos reside en el sistema gestor. Pero las redes seguían creciendo en tamaño y en complejidad, y cada vez se les demandaba mejor rendimiento y disponibilidad, por lo que alrededor de los años noventa los investigadores comprendieron que este diseño de gestión resultaba inadecuado. Gestionar una red muy grande con un único sistema responsable de todas las tareas de gestión no resultaba eficiente, ya que introducía demasiada carga de proceso y de comunicaciones, y demasiada complejidad en dicho sistema. Fue entonces cuando comenzó a abrirse paso el nuevo paradigma de gestión distribuida.

En general, estas primeras aproximaciones hacia la gestión de red distribuida son ahora consideradas como distribución parcial. Las principales tareas de gestión recaen todavía principalmente sobre el gestor principal, y sólo algunas tareas repetitivas y rudimentarias son delegadas a entidades intermedias.

1.3 MODELO DE GESTIÓN EN EQUIPOS DE ESCRITORIO

A finales de los 80 empiezan a proliferar los computadores personales o PCs conectados en redes de área local. El DMTF (entonces Desktop Management Task Force, Grupo de Trabajo de Gestión de Equipos de Escritorio, actual Distributed Management Task Force, Grupo de Trabajo de Gestión Distribuida) trata de salvar este problema, realizando una gestión integrada, independiente del sistema operativo y protocolos de red. Para ello define DMI [6] (Desktop Management Interface, Interfaz de Gestión de Equipos de Escritorio), pensado como un complemento a SNMP. DMI utiliza MIF (Managed Information Format, Formato de la Información Gestionada) como lenguaje de definición de información, y especifica el conjunto de información útil para la gestión de PCs en la denominada Master MIF. Posteriormente, y debido a que gran parte de la información se maneja a nivel de la BIOS (Basic Input Output System, Sistema Básico de Entrada y Salida), trata también de normalizar este acceso con SMBIOS (System Management BIOS, BIOS de Gestión de Sistemas).

1.3.1 DMI [6]

Define un marco normalizado para gestionar y rastrear componentes en un computador de personal, portátil o servidor, siendo el DMI el primer estándar referido específicamente a este fin. Para ello, especifica la arquitectura mostrada en la Figura 1.1 En ella, la capa de servicio de gestión que se encarga de comunicar la interfaz de gestión a la que acceden las aplicaciones gestoras con la interfaz de los componentes a la que se conectan los recursos gestionados. Las operaciones posibles que incluye, se refieren a la obtención, modificación, registro y notificación de información, así como la creación y borrado de ejemplares de la MIF. A partir de la segunda versión de DMI se hace uso de RPCs (Remote Procedure Calls, Llamadas a Procedimientos Remotos) confirmadas, permitiendo también la gestión remota de dispositivos y un marco para el desarrollo de software independiente de la plataforma de gestión.

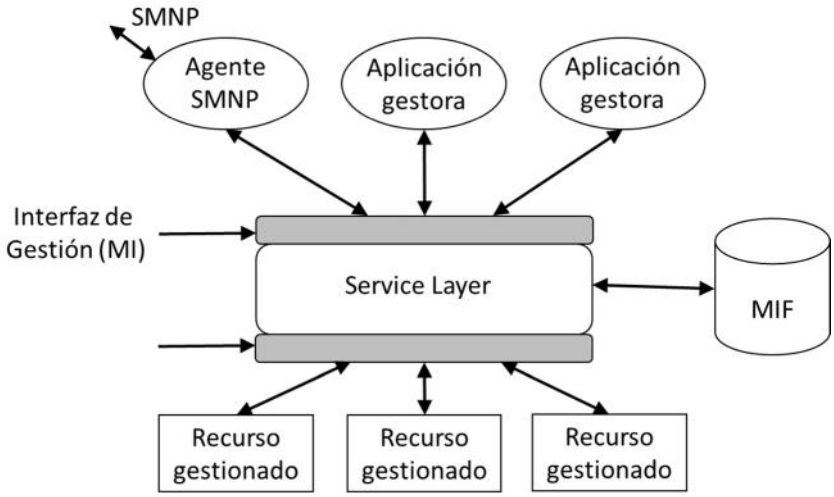


Figura 1.1 Arquitectura DMI [6]

1.3.2 SMBIOS [7]

Aborda cómo los vendedores de equipos y tarjetas pueden presentar la información de gestión sobre sus productos con un formato estándar extendiendo la interfaz de la BIOS. La especificación está pensada para permitir la entrega de esta información con una instrumentación genérica a aplicaciones de gestión que usen DMI o tengan acceso directo, eliminando la posibilidad de operaciones que produzcan fallos, como el sondeo de sistemas para detectar la presencia de hardware.

El lenguaje empleado en DMI [6] se ha usado para definir información relacionada con ordenadores de escritorio. Su funcionamiento es en cierta manera similar al de SMI: sólo se pueden definir componentes, que poseen grupos de variables simples (llamadas en este caso atributos) y tablas. Sin embargo, MIF es un lenguaje incluso más simple que SMI puesto que las claves de cada tabla son siempre internas a dicha tabla, y por lo tanto, no se pueden definir asociaciones a partir de dichas claves. Para identificar los atributos en este caso no existe un árbol de identificadores de objeto, sino que la identificación se basa en el par {clase de grupo, identificador de atributo}, donde la clase de grupo es una cadena que identifica unívocamente a un grupo de atributos, y el identificador es un número entero. Sin embargo, cada grupo también puede tener asignado un OID debajo de la rama enterprises DMTF para permitir el acceso desde SNMP.

Las cláusulas especificadas en MIF son:

- Componente (component). Un componente reúne distintos grupos de atributos, caminos, enumeraciones y tablas. Además, posee un nombre, una descripción

y una cláusula pragma, que puede indicar el OID SNMP con que se corresponde, dependencias con otros grupos, implementación, y registro de Windows.

- Camino (path). Un camino posee un nombre e indica la localización de los archivos que se usan para la gestión del componente en distintos sistemas operativos (DOS, MacOS, OS/2, Unix y distintas versiones de Windows).

- Grupo (group). Un grupo de atributos se define con un nombre, una clase que indica el organismo que define la información, el nombre de lo definido y la versión; una descripción; un identificador numérico del grupo dentro del pragma.

- Atributo (attribute). Un atributo incluye un nombre, una descripción, un identificador numérico del atributo dentro del grupo, el tipo de acceso, el tipo de almacenamiento, el valor que posee y una cláusula pragma. El tipo de datos puede ser un número entero (integer (o int), integer64 (o int64), gauge, counter, counter64), string (n) o displaystring(n), octetstring(n) y date. También puede ser una enumeración.

- Enumeración (enum). Una enumeración indica la relación entre números enteros y cadenas de caracteres.

- Tabla (table). Define ejemplares de tabla dentro de un grupo de atributos. Para ello indica el nombre, identificador y clase, así como los valores de las distintas filas y columnas de la tabla.

1.3.3 Master MIF

La Master MIF [8] contiene todos los grupos de información estandarizados por el DMTF. Dichos grupos se refieren sobre todo a datos de configuración, permitiendo hacer gestión de inventario respecto a los recursos físicos (discos, memoria, tarjetas de red, procesador…) y lógicos (programas instalados) que posee. Una gran parte de esta información se podría corresponder con la HOST-RESOURCES-MIB, si bien se define de manera totalmente independiente. La información que se refiere a los recursos físicos normalmente se obtiene a partir de SMBIOS, ocurriendo que la mayor parte de las BIOS que vienen incorporadas en los PCs actuales implementan esta función.

1.4 MODELO DE GESTIÓN EN PLATAFORMAS DE PROCESAMIENTO DISTRIBUIDO

La tecnología de las plataformas de procesamiento distribuido orientado a objetos. El acceso a la información de gestión mantenida por los recursos gestionados se puede realizar utilizando los mecanismos de comunicación propios de la plataforma de procesamiento distribuido [9]. De esa manera se dispondría de un protocolo de intercambio de información de gestión con una interfaz de acceso a sus servicios estándar que se puede aplicar para acceder directamente a los parámetros gestionables de recursos tales como aplicaciones distribuidas que proporcionan un servicio [10].

1.4.1 CORBA

CORBA [11] es una implementación conforme a ODP (Open Distributed Processing, Procesamiento Distribuido y Abierto) [12] que propone una solución para el modelo de ingeniería y notaciones y mecanismos para construir modelos computacionales. En ella existe un ORB (Object Request Broker, Intermediario de Peticiones a Objetos) que se encarga de poner en comunicación un cliente y un conjunto de objetos. De esta manera, si un cliente desea comunicarse con un objeto concreto, puede acceder al repositorio de interfaces para conocer las operaciones y atributos que implementa.

La arquitectura CORBA (Common Object Request Broker Architecture, Arquitectura Común de un Intermediario de Peticiones a Objetos), ha sido propuesta a través de su Grupo de Trabajo de Telecomunicaciones (TTF, Telecom Task Force) [11]. Éstas se refieren tanto al intercambio de la información como a la definición de la misma utilizando IDL (Interface Description Language, Lenguaje de Descripción de Interfaces).

CORBA define IIOP (Internet Inter-ORB Protocol, Protocolo Inter-ORB de Internet) [12] como protocolo para las comunicaciones entre objetos distribuidos, pudiéndose acceder y modificar valores de atributos, así como solicitar la ejecución de métodos que devuelven resultados o lanzan excepciones.

Al mismo tiempo CORBA define un conjunto de servicios conocidos como CORBAservices (Servicios CORBA) que incluyen entre otros un servicio de nombres [13] para la localización de ejemplares y un servicio de notificaciones [14]. El TTF ha definido servicios adicionales como el de registro [15]. Para el caso de TMN, la recomendación Q.816 [16] utiliza un conjunto de servicios que incluye al de nombres y al de notificaciones y el de registro, así como otros servicios nuevos que permiten operar de manera similar a CMIP. Éstos son, por ejemplo, el localizador de fábricas (para crear ejemplares), de terminación (para borrar ejemplares),

operación de múltiples objetos (que permite realizar operaciones dentro de un ámbito o con filtrado) o localizador de canal (para encontrar canales de eventos).

1.4.2 IDL

IDL [17] es el lenguaje utilizado en CORBA para definir interfaces. Es un modelo orientado a objetos en el que las clases son interfaces con atributos, operaciones y relaciones de herencia. A su vez, se pueden definir asociaciones mediante el uso de referencias a otras interfaces. Las últimas versiones también incluyen características para definir componentes.

La recomendación M.3020 [18] define una metodología de definición de interfaces denominada UTRAD (Unified TMN Requirements, Analysis and Design, Requisitos, Análisis y Diseño de TMN Unificados) que indica que a partir de su diseño en UML (Unified Modeling Language, Lenguaje de Modelado Unificado) [OMG01b] se deben poder generar definiciones tanto en GDMO como en IDL.

IDL no es un lenguaje concebido para definir información de gestión sino para describir las interfaces de acceso que permiten la comunicación entre aplicaciones distribuidas, que pueden estar implementadas en cualquier lenguaje de programación con el que tenga conexiones. Sí existen, sin embargo, el llamado IOR (Interoperable Object Reference, Referencia Interoperable de Objeto) que se puede obtener a través del servicio de nombres y permite acceder a las interfaces.

Las construcciones de IDL, sin tener en cuenta aquéllas referidas a componentes, está basada en la definición de la gramática de IDL, especificada en [19]. Éstas incluyen:

- Módulo (module). Los módulos poseen un identificador y poseen declaraciones. Dichas declaraciones pueden ser de definición de tipos, constantes, interfaces, tipos de valores o excepciones.

- Interfaz (interface). Las interfaces se definen con un identificador, las interfaces de las que hereda y los modificadores de abstracción (la interfaz no puede tener ejemplares directamente) y local (la interfaz no es remota). Contienen exportaciones, que pueden ser de definición de tipos, constantes, excepciones, atributos y operaciones.

- Atributo (attribute). Los atributos se describen con un identificador y el acceso (lectura y escritura). También pueden lanzar excepciones cuando se obtenga o modifique su valor. Además, poseen un tipo de datos, que puede ser de distintas clases. Los tipos básicos pueden ser números de coma flotante (float, double, long double), enteros con y sin signo (octet, short, long, long long), booleanos,

caracteres (char, wchar), cadenas de caracteres y cualquiera de éstos (tipo any). Otros tipos incluyen las estructuras (struct y union), enumeraciones, secuencias y arrays.

- Operación (operation). Las operaciones se especifican con un identificador, el tipo de datos que devuelve, el modo de respuesta (normal o de sólo ida) y un contexto, que indica qué elementos del contexto del cliente pueden afectar al rendimiento de la solicitud. Puede contener un conjunto de parámetros, y emitir una excepción si se produce algún problema durante su ejecución.

- Parámetro (parameter). Los parámetros de una operación poseen un identificador, un tipo de datos y los modificadores que indican si es un parámetro de entrada o de salida.

- Excepción (exception). Indican que se ha producido una condición excepcional durante la realización de una solicitud. Pueden contener un conjunto de miembros, con identificador y tipo de datos.

- Constante (const). Definen valores constantes, esto es, que no varían, mediante un identificador, un tipo de datos y una expresión que especifica el valor.

- Definición de tipos (typedef). Los tipos se pueden redefinir, añadiendo un identificador al tipo definido.

- Tipos de valores (valuetype). Para permitir el paso de objetos por valor y no por referencia, se pueden definir tipos de valores. Estos valores soportan interfaces, pero también poseen valores de estado, que son atributos con modificadores públicos, y factorías, que representan métodos constructores del valor.

Ante la heterogeneidad de modelos de Gestión de Red integrada se hace necesario establecer mecanismos que posibiliten o faciliten la interoperabilidad entre los distintos dominios, dado que se pueden tener variados entornos en los que algunos de estos modelos deban coexistir, con la problemática de gestión que esto conlleva, así buscar vista unificada de los sistemas gestionados, independientemente del dominio al que pertenezcan, Un modelo notable es el que trata de establecer una arquitectura de interoperabilidad de modelos de gestión es la iniciativa WBEM (Web Based Enterprise Management, Gestión de Empresa Basada en Web) y su modelo de información asociado CIM (Common Information Model, Modelo de Información Común) [20], que permite el acceso a diversos dominios de gestión con una interfaz neutra a todos ellos.

1.5 MODELO DE GESTIÓN BASADA EN WEB

El modelo de gestión basado en Web también conocido por WBEM (Web Based Enterprise Management, Gestión de Empresa Basada en Web) se ha definido a partir de una iniciativa que pretende la interoperabilidad de los modelos anteriormente mencionados, aunque también se puede usar directamente como un modelo de gestión integrada. Como su nombre indica, la idea clave de este modelo es reutilizar las tecnologías de la Web sobre las que existe un gran cuerpo de conocimiento, y que ofrecen buenos resultados: HTTP *(Hyper-Text Transfer Protocol,* Protocolo de Transferencia de Hipertexto) y XML *(eXtensible Markup Language,* Lenguaje de Marcas Extensible) marco para la definición de un lenguaje basado en etiquetas o *tags*. Sin embargo, la mayor aportación de esta iniciativa es la de CIM (Common Information Model, Modelo de Información Común), que se ha definido con el objetivo de poder modelar todos los recursos a gestionar de una empresa. Con dicho modelo se han especificado los esquemas CIM, que definen la información de gestión relacionada con dichos recursos, utilizando el lenguaje conocido como MOF (Managed Object Format, Formato de Objetos Gestionados).

Los elementos de este modelo de gestión son [21]:

- Modelo de información: esquemas CIM
- Protocolo de comunicaciones: emplea HTTP y XML de manera conjunta para el intercambio de información
- Lenguaje de especificación de información de gestión: basado en CIM (Common Information Model, Modelo de Información Común).

1.5.1 Modelo de Información: Esquemas CIM

Los esquemas CIM definen el modelo de información de la gestión WBEM. Al igual que en otros modelos, el DMTF ha realizado una labor de normalización semántica definiendo diversos esquemas CIM que describen conceptos comunes. Estos esquemas constituyen la raíz de una jerarquía de herencia [22]:

- Modelo nuclear o core*:* central para todos los dominios gestionados (se define lo que es un elemento gestionado, un servicio, etc.)

- Modelos comunes: que extienden el anterior y se especializan en cada ámbito concreto (gestión de sistemas, nivel físico, protocolos de comunicaciones, etc.)

- Esquemas de extensión: que extienden los esquemas de dominio con conceptos propios del fabricante. Dentro de estos últimos, la jerarquía puede continuar.

- Para la definición de comportamiento en este modelo de información de gestión, se ha definido el esquema de extensión *Policy Schema*,

1.5.2 Lenguaje: CIM / MOF

La especificación de CIM [20] describe un modelo orientado a objetos a partir de su metamodelo. Este metamodelo se describe mediante un diagrama de clases de UML, y define las construcciones básicas de CIM incluyendo Clases, Propiedades, Métodos, Calificadores, y Esquemas, siendo lo suficientemente expresivo para permitir traducciones sintácticas entre CIM y los otros modelos de información. Sus elementos son [20]:

- Elemento nombrado (Named Element). Cualquiera de los elementos del metaesquema.

- Clase (Class). Colección de ejemplares que soportan el mismo tipo: es decir, las mismas propiedades y métodos. Las clases se pueden organizar en una jerarquía de generalización que representa una relación de subtipos entre clases. En este caso no se soporta herencia múltiple. Las clases tienen propiedades y métodos, que se describen a continuación. Una clase puede participar en asociaciones siendo el destino de una de las referencias que posee la asociación. Las clases también tienen ejemplares.

- Propiedad (Property). Valor usado para caracterizar ejemplares de una clase. Se puede pensar como un par de funciones de obtención y modificación que cuando se aplican a un objeto, devuelven o cambian su estado respectivamente. Las propiedades poseen un tipo de datos.
- Método (Method). Representan el comportamiento relevante de una clase. Se declaran con los parámetros de la operación, y el tipo de datos que devuelve. En el caso de una clase no abstracta puede implicar una implementación.

  Se incluyen asociaciones reflexivas para las propiedades y métodos que permiten su redefinición. Un método puede redefinir un método heredado, lo que implica que cualquier acceso al método heredado resultará en la invocación de la implementación del método redefinido. Una interpretación similar se aplica a la redefinición de propiedades.

- Referencia (Reference). Especialización de una propiedad que define el papel que juega cada objeto en una asociación. Las referencias indican los ejemplares de las clases que participan en una asociación concreta.

- Asociación (Association). Especialización de una clase que contiene dos o más referencias. Representa una relación entre dos o más objetos. Debido a la forma en que se definen las asociaciones, se puede establecer una relación entre clases sin afectar a ninguna de las clases implicadas. Es decir, añadir una

asociación no afecta a la interfaz de dichas clases. Sólo las asociaciones pueden tener referencias. Una asociación no puede ser una subclase de una clase que no sea una asociación y cualquier subclase de una asociación es una asociación. Las asociaciones soportan la provisión de múltiples ejemplares de relaciones para un objeto dado. Por ejemplo, un sistema puede estar relacionado con un gran número de componentes de dicho sistema.

- Disparador (Trigger). Indica el reconocimiento de un cambio de estado de un ejemplar de una clase (como la creación, borrado, actualización, o acceso a un objeto) o la actualización o acceso a una propiedad.

- Indicación (Indication). Objeto que se crea como resultado de un disparador. Debido a que las indicaciones son subtipos de clases, pueden tener propiedades y métodos, y organizarse jerárquicamente.

- Calificador (Qualifier). Se usan para proporcionar características adicionales de Elementos Nombrados (una clase, una propiedad o un método, por ejemplo). Los calificadores proporcionan un mecanismo que hace posible la extensión del metaesquema de una forma limitada y controlada. Es posible añadir nuevos tipos de calificadores definiendo sus nombres, proporcionando así nuevos tipos de meta-datos para procesar la gestión y manipulación de clases, propiedades y otros elementos del metaesquema. El uso estos calificadores que permiten ampliar el metamodelo fácilmente también reduce la formalización del mismo, al ser todos los calificadores opcionales y tener una definición únicamente textual.

- Esquema (Schema). Grupo de clases definidas por una misma entidad. Los esquemas se usan para administración y nombrado de clases. Los nombres de las clases deben ser únicos dentro de sus esquemas. Un esquema contiene ejemplares del conjunto de elementos del metamodelo.

1.5.3 HTTP y XML

WBEM hace uso de las tecnologías de la Web, permitiendo reutilizar herramientas ya existentes. Para adaptar HTTP [23] a las necesidades propias de comunicaciones de gestión, se aprovechan las extensiones de HTTP [24] que permiten incluir nuevas cabeceras específicas para una aplicación concreta [8], que indican al servidor que la solicitud que se está realizando es relativa a operaciones de gestión, pudiendo también el servidor devolver respuestas a dichas operaciones. Tras las cabeceras es necesario especificar el tipo de operación que se realiza, y sobre qué información se realiza. Para ello, servidor y cliente intercambian mensajes expresados en XML [25], definiéndose [8] la forma en que se codifican dichos mensajes, a partir de un DTD (Document Type Definition, Definición de Tipos de

Documento) conocido como CIM-XML.

El modelo WBEM utiliza el protocolo de comunicación CIM-XML como protocolo de intercambio de información de gestión [26]. CIM-XML tiene los siguientes componentes:
1. Modelo de información: *Common Information Model* (CIM)
2. Codificación *xmlCIM*
3. Conjunto de operaciones para obtener y manipular datos CIM
4. Encapsulado http

CIM-XML utiliza xmlCIM para codificar la información de gestión y el protocolo http para su transporte. Las extensiones definidas en [20] permiten definir nuevas cabeceras específicas para los servicios o primitivas de gestión. Estas primitivas se codifican mediante etiquetas XML, y se refleja en un DTD Document Type Definition, Definición de Tipos de Documento) conocido como CIM-XML. CIM-XML define todas las interacciones entre gestores y agentes como mensajes CIM, que son paquetes de datos de petición o respuesta utilizados para intercambiar información. Estas operaciones de gestión se transportan sobre HTTP en forma de mensajes de petición/respuesta.

Por otro lado, además de las operaciones es necesario definir la sintaxis en que se transmitirán los datos y metadatos contenidos en los mensajes correspondientes. Dicha codificación se define en [27], representándose en XML la información CIM a partir del mismo DTD que se comentaba anteriormente.
Con todo, la arquitectura de protocolos que se utiliza podría representarse como en la Figura 1.2 [21].

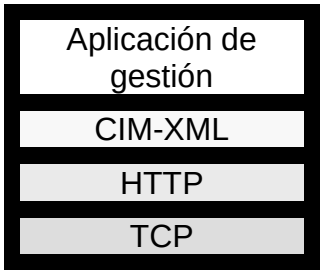


Figura 1.2. Arquitectura de protocolos de HTTP-XML

Las operaciones o métodos pueden invocarse en relación a diferentes tipos de información:

- Sobre datos de gestión propiamente dichos, los ejemplares (modelo de información orientado a objetos): obtención, eliminación, creación, modificación y enumeración de ejemplares, y obtención y modificación de sus propiedades.

Tabla 1.2 Modelos de Gestión de Red Integrada.

Modelo	**Protocolo de intercambio**	**Lenguaje de definición de información**	**Modelos de información**
TMN/OSI-SM	CMIP	GDMO	X.721, M.3100
Internet	SNMP	SMI	MIB-II, otras
Equipos de Escritorio	DMI	MIF	Master MIF
Plataformas de procesamiento distribuido	CORBA/IIOP	IDL	X.780, M.3120
Gestión basada en Web	HTTP-XML	MOF/CIM	Esquemas CIM

- Sobre metadatos, entendiendo por metadatos información de naturaleza más estática (las clases). Son similares a las anteriores, pero ahora sobre clases.

- Consultas: obtener ejemplares que cumplan una condición o restricción. Junto con las operaciones sobre metadatos, suponen una novedad frente a los otros modelos.

- Además CIM soporta otro tipo de operaciones “extrínsecas” que pueden ser o no ser implementadas por los agentes. Estas operaciones se denominan invocaciones de métodos extrínsecos”, y permiten solicitar la ejecución de un método de una clase sobre determinada instancia de esa clase, y, en su caso, devolver el resultado.

Cada una de estas operaciones tendrá su consiguiente mensaje de respuesta. Los recursos gestionados, es decir, los elementos del modelo de información de gestión, también se representan mediante XML.

1.6 ASPECTOS CONCLUYENTES

A lo largo de diversas secciones este capítulo ha presentado las distintas propuestas y aproximaciones que existen para establecer mecanismos de

interoperabilidad entre los distintos modelos de información de gestión. Éstos se resumen en la siguiente tabla:

Por tanto, existen múltiples modelos incompatibles que a su vez poseen lenguajes de definición con distinto grado de expresividad. Además, estos lenguajes son poco formales y con poca o nula capacidad para definir el comportamiento del gestor, lo que los hace difícil de implantar en gestores inteligentes

1.7 OTROS MODELOS PROPUESTOS

La heterogeneidad de los recursos actuales en las redes y servicios de telecomunicaciones ha hecho que se definan diversos modelos de gestión integrada, los cuales generan un sistema aún más complejo que el que intenta resolver, creando un mayor trabajo al administrador de red que tiene que analizar diversos mecanismos y lenguajes para poder hacer una gestión integrada de la red.

Para solventar este dilema se han presentado variadas propuestas para allanar este inconveniente, así encontramos que el primer desarrollo [58], en el cual se presenta un modelo de gestión integrada, en el cual el gestor principal interactúa con los agentes de los diferentes modelos, de esta forma integrar todos los modelos de gestión en un solo punto, a través de generar modelos de correspondencia entre el gestor y cada uno de los Modelos referidos ver figura 1.3 [58],

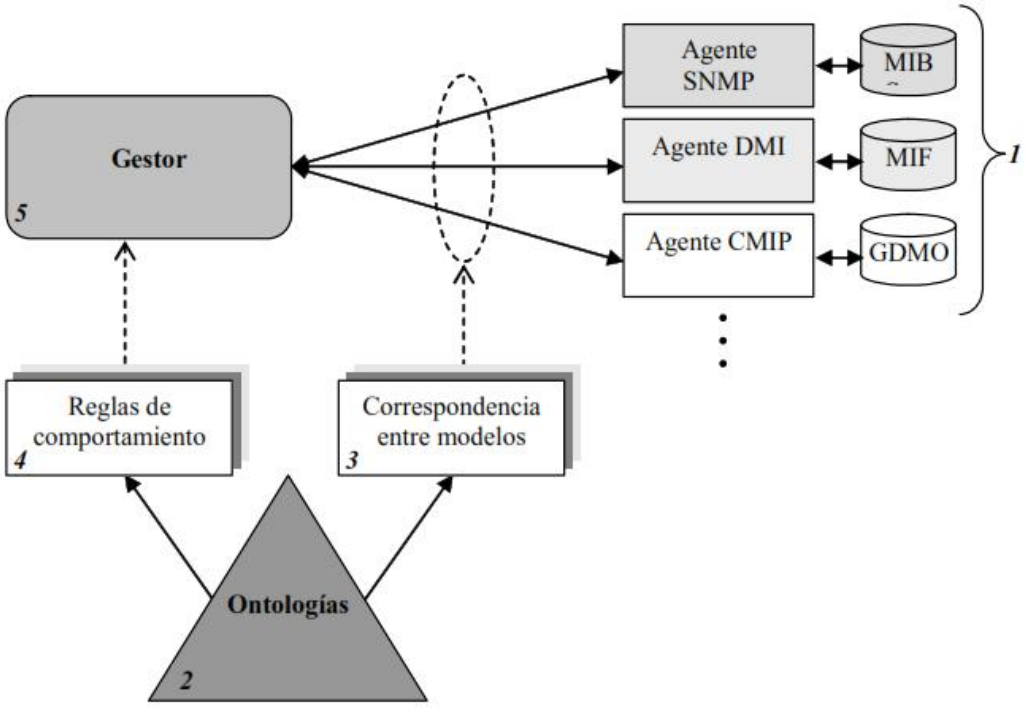


Figura 1.3 arquitectura para un sistema de gestión basado en ontologías [58],

Otra propuesta de modelo de gestión integrada, la podemos ver en [59], la cual presenta un enfoque muy parecido, mejorando las capacidades de gestión sobre los protocolos de los niveles altos de cada sistema de gestión, en la figura 1.4 se puede observar su modelo propuesto denominado de Gestión integrada de red utilizando sistemas multiagentes y basados en Ontologías, que permite a un gestor trabajar con un único modelo de información que, teniendo en cuenta los aspectos semánticos, fusiona las distintas definiciones de recursos gestionados.
.

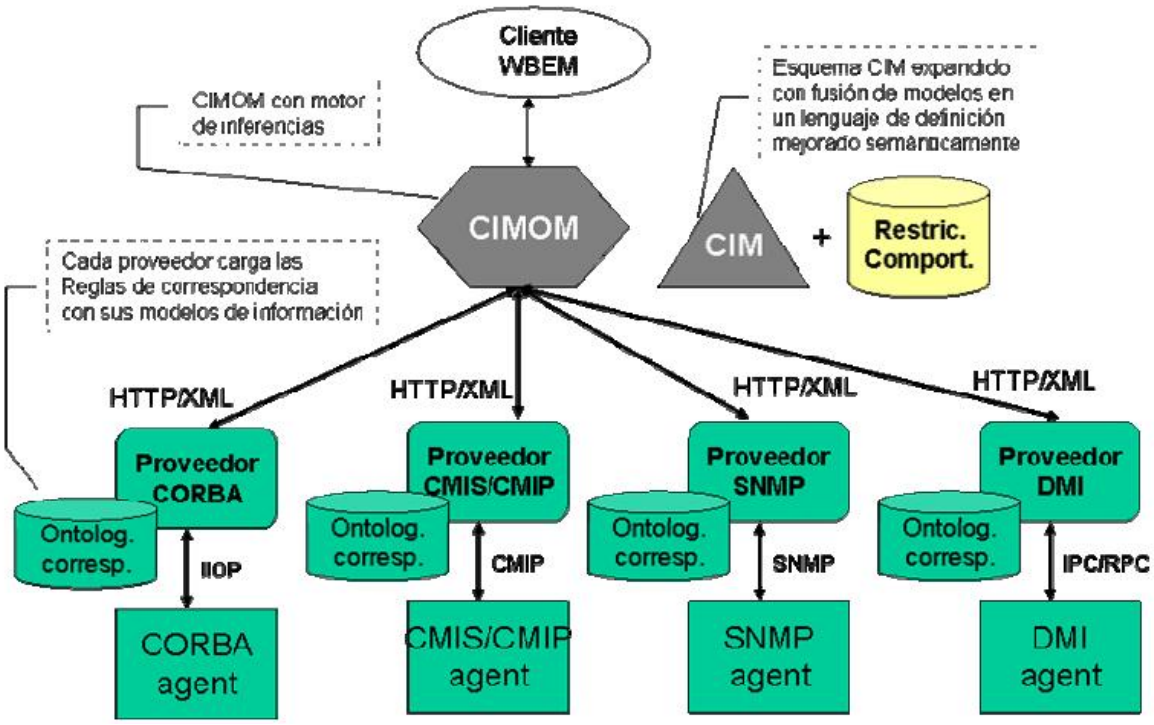


Figura 1.4 Arquitectura propuesta para un gestor de red integrado basado en Ontologías [59]

Partiendo de estos elementos buscamos el desarrollo de nuestro sistema de gestión integrado, inicialmente definimos que significa gestión de red: Consiste en la ejecución del conjunto de funciones requeridas para controlar, planear, asignar, desplegar, coordinar y monitorizar los recursos de una red de telecomunicación, incluyendo funciones encaminamiento de tráfico, gestión de configuración, gestión de fallos, gestión de seguridad y gestión de rendimiento, siguiendo el modelo FCAPS (Fault, Configuration, Accounting, Performance and Security, Fallos, Configuración, Contabilidad, Rendimiento y Seguridad) [60]

Con la definición anterior la Gestión de Red es Integrada surge como consecuencia de la necesidad de gestionar equipos de telecomunicación heterogéneos, y consiste en la posibilidad de gestionar diferentes tipos de recursos de red (recursos de transmisión y conmutación) procedentes de diversos fabricantes utilizando un mismo conjunto de herramientas de gestión. Para ello, la gestión integrada usa conceptos estandarizados de bases de datos de gestión globales; permite una aproximación integral a distintos aspectos, incluyendo los organizacionales, soporte de sistemas heterogéneos; y ofrece programación abierta e interfaces de usuario [61]

2 ONTOLOGÍAS TÉCNICAS DE REPRESENTACIÓN DE CONOCIMIENTO

Ontología como término, tomado de la Filosofía, es la parte de la metafísica que trata del ser en general y de sus propiedades trascendentales. Su definición en el campo de la Inteligencia Artificial es algo difusa, puesto que los expertos en el tema no han acordado una definición conjunta y existen múltiples referencias con definiciones complementarias. Se puede acercar la definición como [28]: una ontología como una especificación explícita y formal de una conceptualización compartida. Esta definición se puede entender de la siguiente manera:

- Es explícita porque define los conceptos, propiedades, relaciones, funciones, axiomas y restricciones que la componen.
- Es formal porque es interpretable por máquinas.
- Es una conceptualización porque es un modelo abstracto y vista simplificada de fenómenos del dominio que se quiere representar.
- Finalmente, es compartida porque la información ha sido consensuada previamente entre distintos grupos de expertos.

Las ontologías se crearon para compartir y reutilizar el conocimiento [28]. Con ello se trataban de solventar los siguientes impedimentos encontrados a la hora de que interoperaran sistemas inteligentes:

1. Representaciones heterogéneas. Existen múltiples aproximaciones para representar el conocimiento, no pudiéndose representar siempre en un formalismo el conocimiento que está representado con otro formalismo.

2. Dialectos en familias de lenguajes. Dentro de una misma familia de formalismos de representación del conocimiento, puede ser complicado compartir conocimiento entre dialectos.

3. Falta de convenciones de comunicación. No existe un protocolo que especifique cómo distintos sistemas pueden consultarse el conocimiento que poseen. En este punto se ha trabajado mucho en el campo de la gestión de red.
4. Desemparejamiento de modelos en el nivel de conocimiento. Aunque los problemas del nivel de lenguaje se resuelvan, sigue siendo difícil combinar dos bases de conocimiento. Estas barreras aparecen cuando se usan diferentes términos primitivos para organizarlas.

Lo interesante es que la ontología debe ser compartida, es decir, resultado de un consenso en una determinada comunidad. Normalmente, el desarrollo de una ontología es un proceso en el que colaboran distintas personas. A veces se dice

que una ontología es un “vocabulario compartido”, pero no son sólo los términos los que se comparten, sino la conceptuación que representan [29].

De esta forma, mediante ontologías, distintos agentes inteligentes podrían compartir y reutilizar conocimiento, intentando resolver problemas de heterogeneidad que se plantean en los modelos iniciales a distintos niveles.

La utilización de la representación mediante ontologías ha ganado enorme relevancia con la aparición de la Web Semántica, cuyo objetivo principal es organizar la información Web y formalizarla, de modo que ésta sea interpretable de forma inteligente por aplicaciones capaces de extraer su significado. Algunas características de la representación mediante ontologías son [30]:

- La ontología más típica para su uso en la Web, tiene una taxonomía, y un conjunto de reglas de inferencia.

- La taxonomía define clases, subclases, y las relaciones entre entidades son una poderosa herramienta para su uso en la Web. Permiten expresar un gran número de relaciones entre entidades asignando propiedades a las clases y permitiendo que las subclases hereden dichas propiedades.

- Las reglas de inferencia añaden capacidad adicional. Una ontología puede expresar la regla.

2.1 LENGUAJES DE DEFINICIÓN DE ONTOLOGÍAS

Existen una variada cantidad de lenguajes dirigidos a la definición de ontologías. Se definen según los patrones de especificación de información y la terminología empleada, Se puede agrupar en disímiles paradigmas. Se habla de redes semánticas, representación basada en marcos, lógica descriptiva, orientaado a objetos, etc. Las diferencias entre unos y otros están en los elementos que incorporan y en el uso que se tiene sobre ellos [31].

Los elementos más comunes [31] son los conceptos o clases, los ejemplares de estas clases, las relaciones que se pueden establecer entre las clases (herencia, asociación, etc.), las propiedades de los ejemplares y las facetas de las propiedades (cardinalidad, cuantificación universal, existencia, etc.).

En cuanto al concepto de semántica formal de los lenguajes o metamodelos, puede decirse que aunque los lenguajes formales tienen una sintaxis formal que pueden comprender las máquinas, no todos disponen de una semántica

formal también interpretable por éstas. Los beneficios de utilizar una semántica formal son [32]:

1. Definición y clarificación precisa de la semántica del lenguaje.

2. Posibilidad de razonar computacionalmente con los elementos de información y extraer conclusiones, lo que permite:
   - Inferir nueva información no establecida explícitamente
   - Comprobar la consistencia de los modelos de conocimiento definidos, es decir, que los elementos de información son coherentes con las restricciones
   - Verificar la corrección de los modelos de información, es decir, que el uso que se hace del lenguaje es coherente con su semántica
   - Y todo ello, con el apoyo de herramientas que pueden interpretar de forma rigurosa el significado de los modelos de información definidos.

Una forma de clasificar la ontología es en pesadas y ligeras. Las ontologías ligeras no incluyen axiomas ni restricciones, por lo que no se puede hacer razonamientos deductivos. Las pesadas sí, por lo que permiten realizar consultas avanzadas, detectar inconsistencias de información e inferir nueva información.
Se pueden establecer distintos tipos de ontologías atendiendo a diversos aspectos como se indica en [33]. Según el ámbito del conocimiento al que se apliquen:

- Ontologías generales: son las ontologías de nivel más alto ya que describen conceptos generales (espacio, tiempo, materia, objeto, etc.)

- Ontologías de dominio: describen el vocabulario de un dominio concreto del conocimiento.

- Ontologías específicas: son ontologías especializadas que describen los conceptos para un campo limitado del conocimiento o una aplicación concreta.

Según el tipo de agente al que vayan destinadas:
- Ontologías lingüísticas: se vinculan a aspectos lingüísticos, esto es, a aspectos gramáticos, semánticos y sintácticos destinados a su utilización por los seres humanos.

- Ontologías no lingüísticas: destinadas a ser utilizadas por robots y agentes inteligentes.

- Ontologías mixtas: combinan las características de las anteriores.

Según el grado o nivel de abstracción y razonamiento lógico que permitan:

- Ontologías descriptivas: incluyen descripciones, taxonomías de conceptos, relaciones entre los conceptos y propiedades, pero no permiten inferencias lógicas.

- Ontologías lógicas: permiten inferencias lógicas mediante la utilización de una serie de componentes como la inclusión de axiomas, etc.

2.2 LENGUAJES DE LA WEB SEMÁNTICA

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida [34]. En el entorno de la Web Semántica [30] existen numerosos grupos de investigación estudiando las ventajas de trabajar con lenguajes de ontologías, su aplicabilidad, límites, problemas, etc. y se han desarrollado numerosas herramientas para la edición de ontologías, validación (comprobación de integridad sintáctica y semántica), inferencia de conocimiento, realización de consultas, etc.

A fecha de hoy, existe un gran camino recorrido en materia de ontologías en la Web Semántica, resultando en diferentes propuestas (member submissions) y estándares (recommendations) por parte del W3C. Estas propuestas constituyen, además, un completo modelo por capas en el que cada esquema de nivel superior se apoya o puede apoyarse en los esquemas de niveles inferiores. En la Fig. 2.1, [30] se presenta un esquema de este modelo de capas con algunos de los lenguajes utilizados en el ámbito de la Web Semántica.

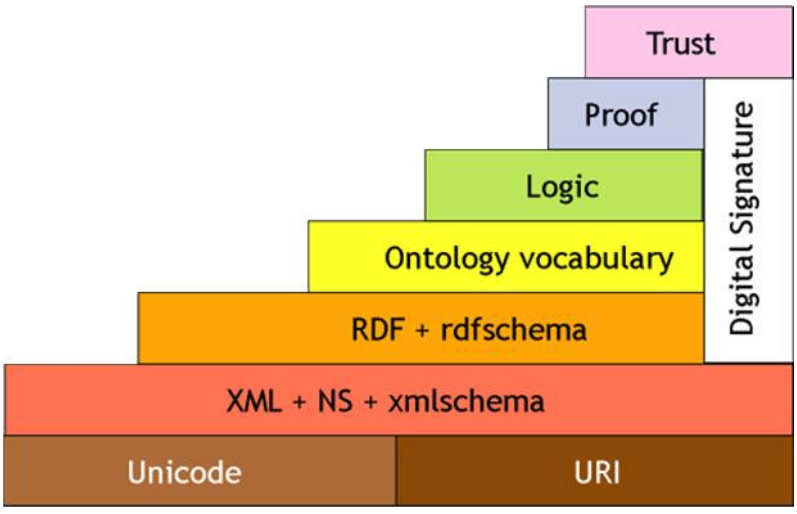


Figura 2.1 Modelo de Capas de la Web Semántica [30]

La Web Semántica es la representación de los datos en la World Wide Web. Es un esfuerzo colaborativo liderado por el W3C con participación de un gran número de investigadores y sociedades industriales.

A continuación se describen brevemente algunos de los lenguajes de la Web Semántica, así como su interrelación.

2.2.1 XML

La base de todos los lenguajes en este ámbito suele ser XML (Extensible Markup Language, Lenguaje de Marcado Extensible). XML proporciona la sintaxis para construir documentos estructurados, pero no impone ninguna restricción semántica. Por ello, no podemos considerarlo un lenguaje de ontologías.

XML es un lenguaje de estructuración sintáctico, constituye la base de este gran modelo. Mediante los espacios de nombres XML (XML namespaces) es posible identificar unívocamente los elementos de los documentos basados en XML. Se basa en la asignación de un espacio de nombres único al documento en cuestión (una URI), y de una referencia relativa para cada elemento única en el contexto de ese documento. Esto permite reutilizar los elementos de los documentos a lo largo de toda la Web.

XML ofrece las siguientes ventajas:

- XML se basa en Unicode, por lo que puede incluir caracteres de otros idiomas aparte del inglés. Esto contrasta con HTML y SGML que se basan en ASCII, lo cual no funciona bien en estos idiomas.
- XML puede estar estructurado. Se puede emplear una DTD para esta estructuración.
- Los documentos XML se pueden construir por composición de otros documentos, empleando para ello métodos de vinculación.
- Puede usarse para contener datos. No se trata de un lenguaje que solamente describa la forma de visualizar los datos del documento (como es el caso del HTML), sino que describe su contenido.
- Proporciona flexibilidad a la hora de definir el grado de detalle del documento XML mediante una DTD.
- Simple de utilizar.

El XML Schema (Esquema XML) es un lenguaje para restringir la estructura de los documentos XML, y también extiende XML con los datatypes (tipos de datos) XSD, útiles para otros estándares [35].

XOL (XML-based Ontology exchange Language, Lenguaje de Intercambio de Ontologías Basado en XML) [36] definido por los autores de OKBC, trata de especificar en XML un subconjunto de las primitivas de este protocolo. Es un lenguaje muy restringido, pues sólo permite definir conceptos, taxonomías y relaciones binarias, y no incluye mecanismos de inferencia, al estar pensado únicamente para el intercambio de ontologías.

2.2.2 RDF

RDF es un marco definido por el Consorcio de la World Wide Web (W3C, World Wide Web Consortium) que incluye un lenguaje [37] con sintaxis XML para integrar una variedad de aplicaciones desde catálogos de bibliotecas y directorios de la Web a la agregación de noticias, software, y colecciones personales de música, fotos y eventos. Las especificaciones del marco RDF proporcionan un sistema de ontología ligera para intercambiar el conocimiento en la Web. RDF define un lenguaje de especificación de ontologías ligero. Es el lenguaje de ontologías sobre el que se ha basado gran parte del cuerpo de conocimiento de la Web Semántica [37].

RDF no proporciona mecanismos para describir propiedades, ni la relación que existe entre ellas y otros recursos, siendo éste el papel de RDFS [39], que describe cómo usar RDF para definir vocabularios, creando para esto un conjunto de términos. Con este lenguaje de descripción de vocabularios se pueden definir clases y propiedades, así como jerarquías y restricciones de tipo.

Es el primer lenguaje de la Web Semántica que incorpora una semántica formal, es decir, que su semántica puede ser comprendida por máquinas [38]. Esta idea es muy importante, y será la que permitirá razonar con los lenguajes para inferir información. Sin embargo, el razonamiento tiene poca aplicabilidad en RDF porque su semántica es muy simple: El elemento básico del RDF es el triple; un recurso (el sujeto) está unido a otro recurso (el objeto, que puede ser una entidad atómica -textos, números...- u otro recurso representado por una URI) mediante un arco etiquetado con otro recurso (el predicado). Se dice entonces que el <*sujeto*> tiene la propiedad <*predicado*> con valor <*objeto*>. Un sujeto puede tener más de un recurso como objeto para el mismo predicado.

RDF necesita de una sintaxis de seriación para hacer disponible los datos en un sistema basado en web. Para ello se escogió el XML. Por tanto, RDF y XML son complementarios, en el sentido de que RDF representa el modelo abstracto mientras que XML proporciona la representación textual concreta del modelo.

El RDF Schema (RDFS, Esquema RDF) [39] es un lenguaje de ontologías que define el vocabulario básico de RDF que debe ser empleado en los triples RDF, en este sentido, es una ampliación del RDF.

Con esta especificación se define una serie de primitivas de modelo adicionales. Entre las más destacadas se encuentran:

- Para añadir a un documento RDF textos legibles para los usuarios humanos y que permitan una mejor interpretación del documento, RDFS proporciona las etiquetas <rdfs:label> (proporciona un nombre del recurso legible para las personas) y <rdfs:comment> (para dar una descripción más larga).
- <rdfs:subPropertyOf> indica que una propiedad está embebida en otra. Por ejemplo tiene-madre sería una subpropiedad de tiene-progenitor.
- <rdfs:seeAlso> y <rdfs:isDefinedBy> indican páginas web relacionadas que contienen información RDF adicional.
- <rdfs:ConstrainResource> y <rdfs:ConstrainProperty> definen mecanismos de restricción avanzados que no son cubiertos por el RDF Schema.
- <rdfs:range> y <rdfs:domain> denen el rango (valores posibles) y dominio (recursos que pueden ser sujeto de esa propiedad) de una propiedad.

Así, la especificación de RDFS puede considerarse como un modelo RDF más, ya que se basa completamente en este lenguaje, pero si tenemos en cuenta la semántica definida para este modelo RDF, entonces éste se convierte en un nuevo lenguaje para especificar otros dominios de conocimiento.

La peculiaridad de este nuevo lenguaje es que los modelos de información basados en él, también pueden hacer uso de RDF, es decir, ambos lenguajes pueden convivir al mismo nivel. Esto se debe a que la semántica de RDFS extiende la de RDF. RDFS permite definir restricciones adicionales sobre los recursos y propiedades de RDF.

2.2.3 OWL

OWL (Ontology Web Language)[40] ha sustituido como estándar del W3C a DAML+OIL en materia de especificación de ontologías. Este último surgió como resultado de los trabajos de OIL, desarrollado en el contexto de un proyecto europeo, y DAML (DARPA Agent Markup Language, Lenguaje de Marcas de Agentes de DARPA), fruto de un proyecto estadounidense. Su sintaxis y semántica son muy parecidos a las de OWL, pero éste último es más completo. OWL se trata de un verdadero lenguaje de especificación de ontologías, entre las ontologías ligeras y pesadas (permite definir ciertos axiomas, pero la expresividad está muy limitada).

El propósito de este lenguaje es similar al de RDFS: proporcionar un vocabulario XML para la definición de clases, sus propiedades y las relaciones entre las clases. Sin embargo, permite al usuario expresar relaciones de mayor riqueza semántica, dotando por tanto de una mayor capacidad de inferencia al procesamiento de un documento. En este sentido, es el paso más avanzado hacia la web semántica. En un esquema de evolución de los lenguajes de marcas:

- XML proporciona una sintaxis superficial para documentos estructurados, pero no impone ninguna restricción de carácter semántico sobre el significado de estos documentos.
- XML Schema es un lenguaje para restringir la estructura de documentos XML. RDF es un modelo de datos para objetos (recursos) y relaciones entre ellos, proporcionando una semántica sencilla para este modelo y pudiendo representarse con la sintaxis XML.
- RDF Schema es un vocabulario para describir propiedades y clases de los recursos RDF, con una semántica para jerarquías de generalización de dichas propiedades y clases.
- DAML+OIL añade más vocabulario para la descripción de propiedades y clases: entre otras, relaciones entre clases, cardinalidad, igualdad, nuevas propiedades y características de propiedades.
- OWL supone un enriquecimiento aún mayor del vocabulario, incluyendo por ejemplo las definiciones de las propiedades simétricas, inversas funcionales. Cuenta con la ventaja de que es el producto de un proceso de revisión más riguroso que el DAML+OIL., OWL es muy similar (salvo un par de construcciones posibles) al DAML+OIL. Esto se explica porque el W3C Web Ontology Group (diseñador del OWL) tomó a DAML+OIL como modelo.

Se han desarrollado tres versiones de OWL con diferentes propósitos:

- OWL Lite: es la versión más reducida, no compatible con todos los documentos RDF/RDFS. Se define como un subconjunto de las construcciones totales existentes para OWL, y además establece restricciones en su uso. Está pensado para principiantes o aquellos que buscan sobre todo la sencillez. Formalmente, su semántica puede considerarse como una extensión de un subconjunto de RDFS.

- OWL DL (OWL Description Logic, Lógica Descriptiva OWL) está destinado a aquellos usuarios que quieren máxima expresividad pero garantizando completitud computacional (posibilidad de llegar a conclusiones basadas en la información existente) e inferencia en tiempo finito. Incluye todas las construcciones definidas para OWL, pero se deben utilizar con ciertas

restricciones para alcanzar las propiedades mencionadas. Esto hace que no sea compatible con documentos que utilizan la máxima expresividad de RDF/RDFS, ya que su semántica es también una extensión de un subconjunto de la de éste.

- OWL Full es la versión más amplia, destinada a aquellos usuarios que quieren máxima expresividad y la libertad sintáctica de RDF. Es compatible con RDF/RDFS, pudiendo concebirse su semántica como una extensión de la de éste, es decir, sigue el mismo procedimiento de construcción sobre RDF y RDFS . Por tanto, los modelos basados en OWL full pueden utilizar libremente construcciones propias de RDF, RDFS y OWL. A cambio de ello, no existen garantías sobre la completitud y el tiempo finito en el razonamiento. Ello se debe a que las propias construcciones que definen el metamodelo pueden utilizarse en los modelos, lo cual significa mucha flexibilidad pero también demasiada libertad.

En la Fig. 2.2, [20], se han representado de forma simbólica los ámbitos de las semánticas de estos lenguajes. Cada elipse representa la semántica de cada lenguaje, y el solapamiento entre elipses implica compatibilidad desde un punto de vista semántico.

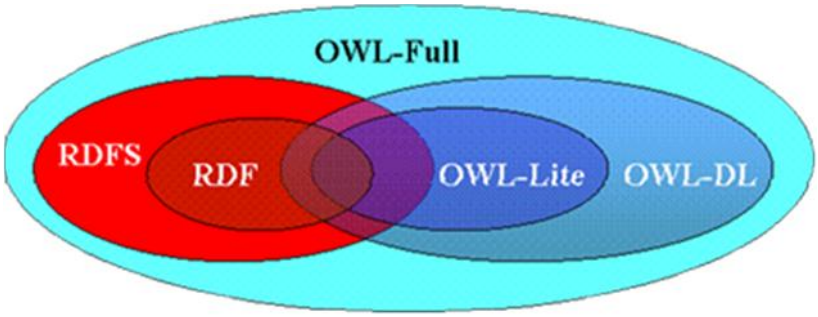


Figura 2.2. Representación del ámbito de las semánticas de los lenguajes de ontologías de la Web Semántica [20]

Las clases OWL están pensadas para poder proporcionar una mayor expresividad que las clases del RDF Schema. Por tanto, OWL define una clase nueva, subclase de rdfs:Class, llamada owl:Class. Esta nueva clase permite, por ejemplo, redefinir el rango de una propiedad en una subclase. Por ejemplo, la clase Animal puede tener una propiedad seAlimentaDe con rango Comida. Por otro lado, una subclase AnimalCarnivoro restringe el rango de seAlimentaDe a Carne. Esto no se puede expresar usando RDFS, ya que rdfs:range impone una restricción global sobre el valor del rango en la clase padre y en todas sus subclases.La sintaxis de intercambio de información más utilizada y la propuesta por la norma es la

sintaxis RDF/XML, aprovechando la compatibilidad (parcial o completa, según el tipo concreto de sublenguaje) existente entre sus semánticas.

Se proponen dos semánticas formales diferentes: Una de ellas es directa, independiente de cualquier otro modelo, basado en la teoría del modelo. Dicha semántica se construye de forma autónoma.

La otra es una extensión de vocabulario de la semántica RDF, que proporciona significado a las ontologías OWL en forma de grafos RDF. Nótese que la semántica de RDF también se basa en la teoría del modelo. De esta semántica se proporcionan dos versiones:

- Una se corresponde de forma más cercana con la semántica directa anterior (semántica para OWL DL).

- La otra puede emplearse en casos donde no existe una separación semántica entre los elementos del metamodelo (clases, individuos, propiedades, etc.). Se trata de la semántica de OWL Full.

Ambas semánticas obligan al soporte de los tipos de datos (datatypes) definidos en XML Schema, aunque también se pueden utilizar otros.

2.2.4 ELEMENTOS DEL LENGUAJE

OWL es un lenguaje de ontologías orientado a objetos, y además OWL DL se encuadra en el paradigma de la lógica descriptiva. Básicamente, define clases, propiedades de las clases y ejemplares de las clases con unos valores determinados de las propiedades (o sin valor), y permite expresar axiomas sobre ellos.

Las características más llamativas son que las propiedades tienen un alcance global, es decir, no son exclusivas de ciertas clases determinadas, y que los tipos de datos son un tipo especial de clases. De esta manera, las propiedades se dividen en dos tipos: aquellas que tienen como valor un tipo de datos (un ejemplar de la clase Datatype) y aquellas que tienen como valor un ejemplar no perteneciente a ningún Datatype.

A continuación se describen los elementos de OWL brevemente. Para ello se agruparán con el objetivo de buscar la claridad y sencillez más que la precisión. Este metamodelo [40] es el mismo para OWL DL y para OWL Full, lo que les distingue es el uso que se hace de él. El de OWL Lite es más reducido.

2.2.5 Elementos referentes a las clases

1. Con respecto a las clases, OWL permite describirlas y establecer axiomas que las relacionan. Las clases OWL se describen en términos de sus extensiones de clase, es decir, el conjunto de ejemplares que pertenecen a dicha clase. Para ello se emplean distintos tipos de descripciones de clase (class descriptions)

2. El identificador de clase, que simplemente la identifica mediante una URI.

3. La enumeración: describe una clase a partir de los ejemplares exactos que definen su extensión, entre los cuales algunos podrán ser equivalentes.

4. Las restricciones de propiedad: definen una clase como el conjunto de ejemplares que cumplen una serie de restricciones en relación a las propiedades:
   - Restricciones de valor: se basan en el valor de las propiedades que presentan los distintos ejemplares. Así, se pueden definir las clases como aquellos ejemplares:
     - Que en caso de tener valor para una propiedad, éste pertenece a una clase determinada, incluyendo los tipo de datos (equivale al calificador de universalidad de la lógica de predicados, es decir, el para todo...).
     - Que deben tener al menos un valor de una clase determinada para una propiedad (calificador de existencia).
     - Que tienen exactamente un valor determinado para una propiedad, incluyendo valores tipados.

   - Restricciones de cardinalidad: se basan en el número de valores que pueden tener las propiedades aplicadas a cada ejemplar. Se pueden indicar cardinalidades máxima, mínima y exacta.

   - Intersección, unión y complemento de otras descripciones de clase. Para entender el complemento, hay que tener en cuenta que se considera que todos los individuos son miembros de la extensión de clase de Thing, mientras que la clase Nothing representa el vacío. Equivalen a los operadores lógicos AND, OR y NOT respectivamente.

OWL DL permite que unas descripciones de clase se aniden dentro de otras, y también las clases anónimas (sin identificador).

Con respecto a los axiomas, existen diversos tipos que modifican o restringen el significado extensivo de las clases (su extensión de clase), no su significado intensivo (el concepto que representan):

- El propio identificador de clase como descripción de clase, que establece simplemente la existencia de una clase.
- La extensión o especialización de una clase, es decir, el mecanismo de herencia.

Éste permite construir modelos de conocimiento cada vez más concretos.

- Equivalencia de extensiones de clases.
- Disyunción de extensiones de clase (aquellas cuya intersección es el conjunto vacío).

2.2.6 Elementos referentes a las propiedades

Las propiedades establecen relaciones entre dos elementos: el sujeto y el objeto. El conjunto de posibles sujetos define el dominio de la propiedad, mientras que el conjunto de posibles objetos define el rango. En OWL DL el dominio siempre debe ser una clase OWL. Según el tipo de clases admitidas como rango, las propiedades pueden ser de tipo ObjectProperty (el rango está formado por clases OWL) y DatatypeProperty (el rango está formado por valores).

Por otro lado, el alcance de las propiedades es global, es decir, en principio su dominio y rango son todo el conjunto de ejemplares (la clase Thing). Mediante los axiomas, ambos se pueden restringir. Esta restricción es global, frente a las restricciones de propiedad que se incluyen en las descripciones de clase, que son locales, ya que sólo afectaban a la clase descrita.

Al igual que con las clases, las propiedades tienen dos significados: el intencional o concepto referido y el extensional. Este último se refiere al conjunto de pares de ejemplares que pueden formar parte del dominio y rango de la propiedad respectivamente.

Los tipos de axiomas son:

- Axiomas basados en construcciones provenientes de RDF Schema:
  - Especialización: establece que una propiedad es subpropiedad de otra, es decir, si un par de ejemplares verifican una propiedad, entonces también verifican la otra. Ambas propiedades deben ser del mismo tipo, la primera sería la especializada y la segunda la más general.
  - Dominio y rango de la propiedad: establecen dominio y rango de una propiedad respectivamente, a partir de una descripción de clase referenciada por su identificador.

- Axiomas para establecer relaciones entre las extensiones de las propiedades:
  - Equivalencia de extensiones.
  - Propiedades inversas: el sujeto de una propiedad es el objeto de otra, y viceversa.

- Restricciones de cardinalidad global: restringen de manera global el número de valores que admiten las propiedades. Distinguimos:
  - Propiedades funcionales (de tipo FunctionalProperty): admiten como máximo un solo valor u objeto para cada sujeto.
  - Propiedades inversa-funcionales (de tipo InverseFunctionalProperty): admiten sólo un sujeto para cada valor objeto, aunque cada sujeto puede tener varios objetos. Sólo pueden ser propiedades de este tipo las propiedades de tipo ObjectProperty.

- Características lógicas de las propiedades:
  - Propiedad transitiva (de tipo TransitiveProperty): verifican que si P(x,y) y P(y,z), entonces P(x,z), donde P es una propiedad y x e y son ejemplares. Un ejemplo sería la propiedad perteneceA en el contexto de regiones geográficas.
  - Propiedad simétrica (de tipo SymmetricProperty): verifican que si P(x,y), entonces P(y,x). Por ejemplo, la propiedad tieneHermano.

2.2.7 Elementos referentes a los Individuos

Los más comunes, y que implícitamente ya hemos supuesto que existen, son:
- Aquellos que establecen la pertenencia de un ejemplar a una clase. Nótese que un ejemplar puede definirse como perteneciente a varias extensiones de clase, lo que fuerza a que dichas extensiones solapen.
- Aquellos que establecen relaciones entre ejemplares (que pueden ser literales tipados para el objeto) mediante propiedades.

En OWL no existe la suposición de nombres únicos, es decir, un mismo individuo puede ser referenciado mediante diferentes nombres. Para lograr esto se emplean los axiomas de identidad de ejemplares:
- Igualdad entre individuos, ahora una igualdad en significado intensivo (concepto que representa la clase).
- Desigualdad entre ejemplares es necesario establecerla explícitamente cuando lo deseamos, ya que no hay nada que a priori impida que dos ejemplares con distinto nombre sean iguales.

Otros elementos interesantes son:

- Los datatypes enumerados: son un conjunto finito de literales tipados, de cualquier tipo.
- Las anotaciones (propiedades de tipo AnnotationProperty): permiten añadir comentarios sobre las clases, propiedades, ejemplares y la propia ontología completa: información de versión, referencia a otro elemento, etiquetas, etc.
- Las anotaciones más específicas de las ontologías completas (propiedades de tipo OntologyProperty): permiten especificar importación de otras ontologías, versiones previas, compatibilidad o incompatibilidad entre versiones, elementos obsoletos, etc. La importación de ontologías es un mecanismo que permite diseños de ontologías modulares y distribuidos.

Existen numerosos beneficios de la aplicación de las ontologías a la especificación de información de gestión: ventajas en cuanto a la expresividad de los lenguajes de ontologías, capacidades para un tratamiento más avanzado de la información, integración de conceptos con semántica común, etc. Mediante la utilización de ontologías es posible especificar axiomas de comportamiento de gestión de forma integrada con la propia información de gestión, gracias a la posibilidad de definir axiomas y restricciones.

3 AGENTES Y SISTEMAS MULTIAGENTE

Este capítulo mostrará los aspectos relativos al concepto de agente y sistema multiagente. No hay una definición universalmente aceptada, por lo que esta se basará en diversas definiciones y características que debiera tener un agente. Estas definiciones y características permiten a continuación una clasificación de agentes según sus características dominantes: móviles y colaborativos

3.1 EL CONCEPTO DE AGENTE

Su origen se sitúa en el año 1977, en el que Carl Hewitt propone la idea de un objeto auto-contenido, interactivo y ejecutado de modo concurrente llamado actor. Este objeto tenía un estado interno con cierto grado de encapsulado y podía responder a mensajes de otros actores [41], es un agente computacional que tiene una dirección de correo y un comportamiento. Los actores se comunican mediante el paso de mensajes y llevan a cabo sus acciones de forma concurrente [74]. Sin embargo, a pesar de lo extendido de su aplicación, su definición es todavía objeto de discusión y análisis. De hecho no existe una definición académica aceptada por todo el mundo.

En una definición más concreta podemos tener: Los agentes autónomos son sistemas computacionales que habitan algún entorno dinámico complejo, perciben y actúan autónomamente en ese entorno y, haciendo esto, llevan a cabo un conjunto de metas o tareas para los que han sido diseñados [42]. Los agentes inteligentes son entidades software que ayudan a la gente y actúan en su nombre:

- Todos los agentes son *autónomos*. Es decir, tienen control sobre sus propias acciones.
- Todos los agentes son *orientados a metas*. Los agentes tienen un propósito y actúan de acuerdo con él.
- Un agente podría también ser *dirigido mediante reglas*, que es un modo más general de definir las metas de los agentes.
- Todos los agentes *reaccionan* ante los cambios detectados en su entorno.
- Algunos agentes son *sociables*. Esto es, interactúan o se comunican con otros agentes.
- Algunos agentes *se adaptan*. Aprenden o cambian su comportamiento basándose en experiencias previas.
- Algunos agentes son *móviles*. Se mueven de máquina en máquina.
- Algunos agentes *se esfuerzan en ser creíbles*. Aparecen ante el usuario como una entidad visible o audible e incluso pueden tener aspectos emotivos o de personalidad[43].

3.2 CARACTERÍSTICAS DE UN AGENTE

La definición anterior deduce una serie de características básicas que debe o puede tener un agente [42].

- **Autonomía.** Es la capacidad del agente de existir independientemente de un usuario o de otra entidad, teniendo el control de sus propias acciones.
- **Reacción ante cambios de su entorno.** Los agentes perciben los cambios producidos en su entorno y responde a esos cambios mientras se adapta a ellos para conseguir su objetivo.
- **Sociabilidad.** Los agentes suelen mantener comunicaciones complejas, ya sean con otros agentes o con humanos, empleando para ello un *lenguaje de comunicación entre agentes* (*Agent Communication Language, ACL*).
- **Pro-actividad.** Un agente no se conforma con responder a los cambios de su entorno o a las acciones ejecutadas directamente sobre él, sino que toma la iniciativa por sí mismo para alcanzar su objetivo.
- **Movilidad.** Es la habilidad del agente para transportarse a sí mismo de una máquina a otra, manteniendo su estado actual.
- **Continuidad temporal.** Los agentes deben ser vistos más como procesos que corren continuamente que como meras funciones que responden con una salida fija ante un mismo valor de la entrada. Mientras dure su ciclo de vida, los agentes continúan ejecutando su código siempre que se produzca el evento adecuado.
- **Credibilidad.** Ningún agente debe proporcionar información falsa de un modo intencionado.
- **Benevolencia.** Los agentes están dispuestos a colaborar mientras esta cooperación no entre en conflicto con sus objetivos a cumplir.
- **Multi-plataforma.** Los agentes deben ser capaces de comunicarse entre diferentes arquitecturas y sistemas de computadores.
- **Robustez.** Los agentes deben ser diseñados para tratar con los cambios inesperados de su entorno. Deben incluir mecanismos de recuperación ante errores del sistema o humanos.
- **Responsabilidad.** Los agentes deben manejar la información privada de una forma responsable y segura.
- **Inteligencia.** Capacidad de aprender y razonar.
- **Capacidad de representación.** Un agente puede actuar en nombre de alguien o algo, esto es, actuar en interés de, en representación de, o en beneficio de alguna entidad.
- **Cooperación.** Capaz de coordinarse con otros agentes para alcanzar un objetivo común.

3.3 TAXONOMÍA DE LOS AGENTES

Clasificamos los agentes basados en varias dimensiones: Respecto a su movilidad los agentes se dividen en *móviles* o *estáticos* [44], según su capacidad o no de desplazarse en una red. Los agentes estáticos son aquéllos que solamente pueden ejecutarse en la máquina donde fueron iniciados. En contraste, un agente móvil es aquél que no se ve limitado al sistema donde comenzó su ejecución, sino que es capaz de transportarse de una máquina a otra a través de la red. Los agentes móviles se emplean, por ejemplo, para situaciones en las que no se puede tener una alta velocidad de comunicación a través de una red y el agente necesita acceder a datos que se encuentran en otro punto de esa red.

En segundo lugar, un agente puede clasificarse como *deliberativo* o *reactivo* [44]. Los primeros derivan del paradigma de pensamiento deliberativo: los agentes poseen un modelo simbólico interno de razonamiento y se ocupan de la planificación y la negociación con el objetivo de conseguir coordinarse con otros agentes.

Los agentes reactivos, por el contrario, no poseen ningún modelo simbólico interno de su entorno y actúan empleando un comportamiento del tipo estímulo-respuesta, respondiendo al estado actual del entorno en que están embebidos.

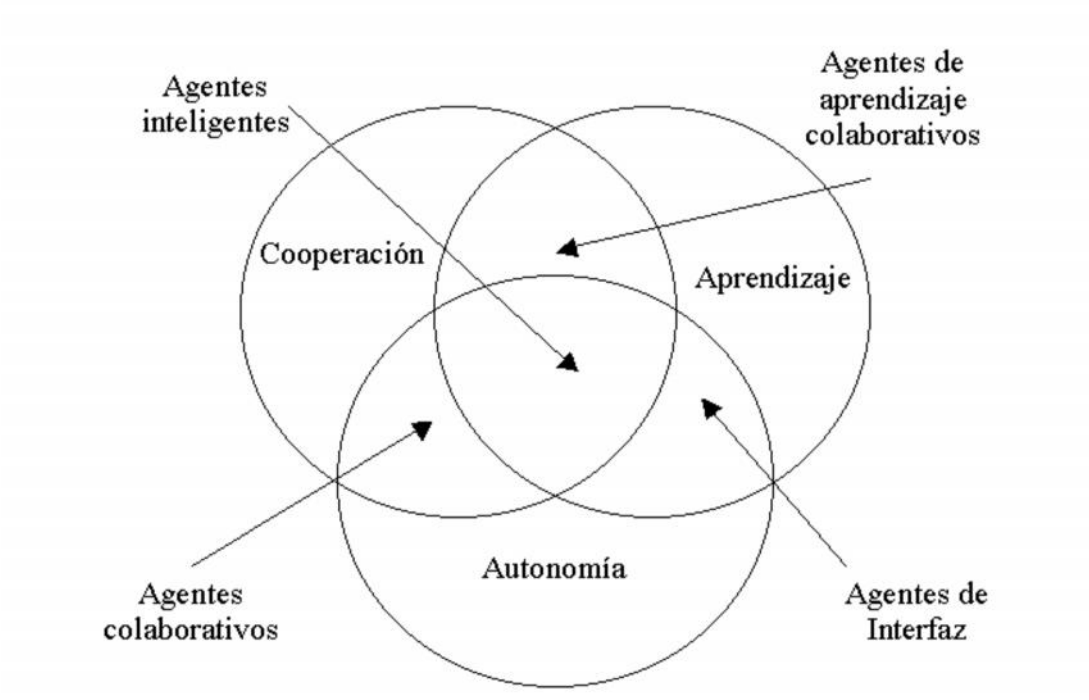


Figura 3.1. Clasificación de los agentes según su característica primaria [44].

En tercer lugar, los agentes pueden clasificarse según los atributos primarios que deberían mostrar. Se distinguen tres principales [44]: autonomía, aprendizaje y cooperación. De este modo existen los agentes *autónomos*, *de aprendizaje* y *cooperativos.* Las combinaciones dos a dos de estas características originan los agentes *colaborativos* (cooperación y autonomía), *colaborativos de aprendizaje* (aprendizaje y cooperación) y *de interfaz* (aprendizaje y autonomía).

Finalmente, la combinación de los tres elementos da como resultado los agentes *inteligentes.* Esta distribución se muestra en la Figura 3.1.

En algunas ocasiones, los agentes pueden ser clasificados por el papel que desempeñan. Por ejemplo, los agentes *de información* o los agentes *de Internet*.

Estos tipos de agentes se dedican a la búsqueda y procesamiento de información en una red, como en el caso de Internet.

Por último, se habla de agentes *híbridos* para referirse a los que combinan dos o más de las categorías anteriores.

Debemos advertir que esta clasificación debe ser considerada como meramente ilustrativa. Los desarrolladores de agentes pueden ajustarse a uno de estos conceptos, o crear una nueva categoría según sus necesidades.

3.4 SISTEMAS MULTIAGENTE

Los sistemas multiagente (*Multi Agent Systems*, MAS) constituyen el siguiente paso lógico en la tecnología de agentes. Por ejemplo el caso de un agente de información que es activado en una red de computadores. Este único agente se movilizará a lo largo de la red tratando de conseguir la información que le interesa a su usuario. En este caso, no existe interacción con otros posibles agentes del entorno.

Son muy pocos los campos en que un único agente fuera capaz de llevar a cabo una tarea compleja, lo correcto es crear una *sociedad* de agentes que se comuniquen entre ellos, colaboren y se coordinen en la realización de la tarea. Estas sociedades son las conocidas como Sistemas Multiagente.

Las principales razones que justifican los sistemas multiagente son [45], [46]:

- Se podría diseñar un único agente que implementase toda la tarea. No obstante, este tipo de *macroagente* representa un cuello de botella para la velocidad, confianza, robustez y mantenimiento del sistema. Dividir la funcionalidad entre varios agentes proporciona modularidad, flexibilidad, facilidad para ser

modificado y extensibilidad. Por tanto, son ideales para la idea del "*divide y vencerás"*. Es decir, el problema se subdivide y se diseña un agente o conjunto de agentes especializados en la resolución de cada uno de los subproblemas.

- El conocimiento que está distribuido entre varias fuentes (agentes) puede ser recombinado en una visión más general cuando sea necesario.

- Las aplicaciones que necesitan computación distribuida son tratadas de una mejor forma por los MAS (Multi Agent System). En esta situación, los agentes pueden ser diseñados como componentes autónomos especializados que actúan en paralelo. Por tanto, los agentes constituyen el nivel más avanzado en cuanto a tecnología de componentes distribuidos. Además estos sistemas pueden estar *abiertos*, es decir, los diversos agentes pueden añadirse o destruirse en el sistema de una forma dinámica.

- Los sistemas multiagente son una plataforma atractiva para la convergencia de varias tecnologías de la Inteligencia Artificial.

- Se evita que un sistema quede bloqueado por un único punto de fallo.

Estas sociedades son más complicadas de diseñar que un sistema de un único agente, puesto que aparte del código necesario, el diseñador del sistema tiene que hacer frente a aspectos relacionados con la comunicación y la negociación entre los agentes, así como de la organización de los agentes dentro del sistema. Sin embargo, la atención de los desarrolladores se centran precisamente en el poder de los MAS, como lo demuestra la gran cantidad de herramientas para la implementación de MAS y el esfuerzo de organizaciones como OMG y FIPA [47] para crear estándares para la interoperabilidad de agentes.

Estos estándares persiguen que sistemas heterogéneos, implementados con diferentes herramientas, diferentes filosofías, con diferentes fines e incluso por diferentes desarrolladores, sean capaces de interactuar, comunicándose en una estructura global compatible. Para ello, y en primer lugar, se debe establecer tanto un lenguaje común (que a partir de ahora llamaremos *lenguaje de comunicación de agentes*, *ACL*) como una organización compatible de los agentes dentro de los MAS que sean ampliamente aceptados.

3.5 LENGUAJES DE COMUNICACIÓN DE AGENTES

Debido a las particularidades de los sistemas multiagente, los aspectos de la comunicación entre agentes cobran un papel de enorme importancia, ya que es casi imposible que exista cooperación sin comunicación. En este sentido, lo que se ha

demostrado como más provechoso es la comunicación a través de *mensajes* ya que aparte de una transmisión de datos, tiene asociado un contenido semántico accesible por ambas partes y sobre el que deben estar de acuerdo. Esto permite pasar de un modelo de objetos a un modelo de agentes inteligentes basados en actos de comunicación.

Los dos lenguajes de comunicación de agentes más extendidos en el desarrollo de agentes son el KQML y el FIPA-ACL [48], ambos basados en la teoría del habla (*Speech Act Theory).* Describiremos brevemente esta teoría antes de pasar al análisis de ambos lenguajes.

3.5.1 La teoría del habla (Speech Act Theory)

La teoría del habla *(Speech Act Theory)* es un marco teórico de alto nivel desarrollado por filósofos y lingüistas para explicar la comunicación humana. Esta teoría ha sido usada, formalizada y extendida con campos de la Lingüística Computacional y la Inteligencia Artificial como un modelo general entre agentes arbitrarios [48]. Se refiere sobre todo al papel del lenguaje como acto comunicativo (*speech act*), de modo que los interlocutores no se limitan solamente a enunciar oraciones que sean verdaderas o falsas. De hecho, un acto comunicativo se compone a su vez de tres actos:

- **locución** El acto físico de la emisión, pronunciación o expresión del acto comunicativo.

- **ilocución** El acto de transmisión de las intenciones del hablante hacia el oyente que se realiza por medio de la emisión.

- **perlocución** Acciones que ocurren como resultado de la ilocución. Por ejemplo, la locución. Cierra la puerta. que realiza un individuo A hacia otro B, lleva consigo la ilocución de un mandato hacia B de cerrar la puerta. Si todo va bien, se producirá la perlocución de que B cierre la puerta.

La ilocución suele dividirse a su vez en dos partes: *la fuerza ilocutiva* (*illucutionary force) y* una *proposición.* La fuerza ilocutiva es el parámetro que permite dividir los actos de comunicación en *afirmativos*, *directivos*, *declarativos, expresivos* y *de compromiso*.

Lo que nos interesa de esta teoría es que constituye la base de los lenguajes de comunicación de agentes. Actualmente existen unos 4600 actos de comunicación catalogados [49].

3.5.2 KQML

El KQML (*Knowledge Query Manipulation Laguange*) ha sido un lenguaje de comunicación entre agentes tomado como estándar de facto durante mucho tiempo. Sus dos especificaciones (una primera versión de 1993 [50], [51]) persiguen crear un lenguaje que permite a los agentes software autónomos y asíncronos compartir su conocimiento y trabajar de modo cooperativo para la resolución de problemas. Fue desarrollado como parte del *ARPA Knowledge Sharing Effort (KSE)*, responsable a su vez de otros lenguajes como el KIF y Ontolingua. Actualmente su uso se encuentra extendido, existiendo numerosos paquetes y herramientas que implementan la comunicación a través de este lenguaje.

El KQML se trata de un lenguaje de alto nivel, orientado a mensajes. Asimismo se define como un protocolo para el intercambio de información *independiente* de la sintaxis, ontología y lenguaje *del contenido* del mensaje. También es independiente del mecanismo de transporte del mensaje (RMI, email.) y de los protocolos de alto nivel involucrados. Cada uno de los agentes posee y gestiona una base de conocimiento virtual (*Virtual Knowledge Base, VKB*), permitiéndose que un agente pueda manipular y realizar búsquedas en la VKB de los otros agentes del sistema. Se basa en una lista de paréntesis balanceados [51], cuyo elemento inicial es el acto de comunicación implicado (llamado *performativa*8). O sea, dota de fuerza ilocutiva al mensaje.

3.5.3 FIPA-ACL

La FIPA (*Foundation for Intelligent Physical Agents*) es una organización internacional sin ánimo de lucro creada en 1995 y con sede en Ginebra (Suiza).
FIPA se dedica a promover la industria de los agentes inteligentes desarrollando especificaciones que soporten la interoperabilidad entre agentes y aplicaciones basadas en ellos. Esto se lleva a cabo mediante la colaboración de sus miembros: compañías y universidades que investigan y desarrollan dentro del campo de los agentes. Los resultados provenientes de estas actividades, son puestas a disposición de todos los desarrolladores mediante una serie de documentos, disponibles a través su página web [47].

Dentro de las numerosas especificaciones FIPA figura la relativa al lenguaje de comunicación de agentes. Este lenguaje, llamado FIPA-ACL (*FIPA Agent Communication Language*), consiste en una evolución del KQML[47].

Un mensaje FIPA ACL se compone de un conjunto de uno o más elementos. De un modo preciso, según la situación varía qué elementos se necesitan para una comunicación efectiva entre agentes. El único elemento obligatorio en todos los

mensajes FIPA ACL es el de performative, aunque sería lógico esperar que la mayoría de los mensajes incluyesen los campos sender, receiver y content.
Podemos, por tanto, distinguir entre el *tipo de mensaje* (determinado por el valor del campo performative) y el de los parámetros del mensaje [47].

1. **Tipo de mensaje en FIPA-ACL: *el campo performative.*** Con este campo, se define el significado principal del mensaje enviado o, lo que es lo mismo, lo que el agente emisor del mensaje pretende con él. Los posibles valores para este campo son los siguientes:
   accept-proposal, agree, cancel, cfp, confirm, disconfirm, failure, inform, inform-if, inform-ref, not-understood, propagate, propose, proxy, query-if, query-ref, refuse, reject-proposal, request, request-when, request-whenever y subscribe.

   Así, por ejemplo, un agente mandaría un mensaje con un valor de la performativa *propose* si lo que desea es enviar a otro agente una propuesta para llevar a cabo una determinada acción dada unas ciertas condiciones previas.

   La especificación FIPA-ACL con respecto de la del lenguaje KQML es su definición formal mediante una lógica modal es mucho más detallada, que elimina cualquier tipo de ambigüedad sobre el verdadero significado del acto de comunicación. A modo de ejemplo, la siguiente definición formal corresponde al acto de comunicación *propose* [52]. Para una descripción de la lógica.

   < *i*; *propose* (*j*;< *i*;*act* >;) >_
   < *i*; *in f orm*(*j*; *I j Done* (< *i*;*act* >;)) *Ii Done* (< *i*;*act* >;)) >
   FP : *Bi* ^:*Bi* (*Bi f j* _ *Ui f j*) *RE* : *Bj*
   donde
   = *Ij Done* (< *i*;*act* >;))*Ii Done* (< *i*;*act* >;)

   Según esta definición formal con un acto de comunicación del tipo *propose*, un agente *i* informa a un agente *j* que, una vez que *j* informe al agente *i* de que *j* ha adoptado la intención de que *i* lleve a cabo la acción *action*, y de que las condiciones previas para que *i* ejecute la acción hayan sido establecidas, *i* adoptará la intención de llevar a cabo la acción *act*.

   Los campos FP y RE hacen referencia respectivamente a las *Condiciones Previas de Viabilidad* (*Feasibility Preconditions*) y a los *Efectos Racionales* (*Rational Effects).* Las FPs de un acto comunicativo son las condiciones que deben cumplirse antes de que un agente pueda llevar a cabo una acción. Por otro lado, un RE es el efecto que un agente espera producir con la ejecución de la acción. En este sentido, conviene aclarar que el agente no se ve atado en ningún momento al cumplimiento de los REs asociados al acto de comunicación.

Es decir, puede emplear los REs junto a su conocimiento para poder cumplir con sus objetivos, pero no preocuparse de si dichos REs se han producido.

2. **Parámetros de un mensaje FIPA-ACL:** Los parámetros básicos de un mensaje FIPA-ACL se pueden dividir en las siguientes categorías [53]:

3. **Relativos a los participantes en la comunicación:**
   _ sender: Indica la identidad del emisor del mensaje. Tiene asociada la dirección de un agente. Es un elemento que aparece en la gran mayoría de los mensajes FIPA-ACL, aunque es posible omitirlo, por ejemplo, cuando el emisor desea permanecer en el anonimato.

   _ receiver: Indica la identidad del receptor del mensaje. Tiene asociada la dirección de un agente. De modo general, este campo debe aparecer en todos los mensajes. Solamente está permitida su omisión si el receptor del mensaje puede ser deducido del contexto o, en casos especiales, a partir de un mensaje embebido en el contenido del mensaje.

   _ reply-to: Este elemento indica que los mensajes posteriores del presente flujo de conversación tienen que ser dirigidos al agente que aparece en este campo, en vez del agente que aparece en el campo sender.

4. **Relativos al contenido del mensaje.**
   _ content: Indica el contenido del mensaje. De modo equivalente denota al objeto de la acción. La mayoría de los mensajes FIPA-ACL necesitan de este parámetro, aunque algunos mensajes como los del acto de comunicación cancel, tienen un contenido implícito.

5. **Relativos a la descripción del contenido del mensaje.**
   _ language: Indica el lenguaje en que se expresa el contenido del mensaje. Este campo puede omitirse si se asume que el agente receptor, conoce el lenguaje en que está expresado el contenido. El estándar en la versión de 1999 define un lenguaje de contenidos de carácter general llamado SL así como tres subconjuntos del mismo denotados como SL0, SL1 y SL2.

   _ encoding: Indica la codificación específica del lenguaje en que está expresado el contenido del mensaje.

   _ ontology: Indica la(s) ontología(s) empleadas para dar significado a los símbolos en la expresión del contenido del mensaje. Se emplean las ontologías conjuntamente con el elemento de lenguaje para dar soporte a la interpretación del contenido por parte del agente receptor. Suele omitirse si se da por entendido que los agentes conocen previamente el valor de este campo.

6. **Relativos al control de la conversación.**

_ protocol: Indican el protocolo de interacción que el emisor del mensaje está empleando en el mensaje actual. Este elemento es opcional, aunque se recomienda su uso para evitar problemas de ambigüedad al emplear directamente la semántica ACL para el control de la generación e interpretación de los mensajes.

_ conversation-id: Introduce un identificador de conversación que se emplea para identificar la secuencia actual de actos de comunicación que forman dicha conversación. El empleo de este campo permite el seguimiento y análisis de las conversaciones por parte de los agentes.

_ reply-with: Introduce una expresión que será empleada por el agente que responda al mensaje actual para hacer referencia al mismo. Así un agente que reciba un mensaje con la expresión

reply-with <expr>
responderá con
in-reply-to <expr>

_ in-reply-to: Denota una expresión que referencia una acción anterior del que el presente mensaje es una respuesta.

_ reply-by: Denota una expresión relativa a un tiempo y/o fecha que indica el límite temporal en el que el emisor del mensaje quisiese haber recibido una respuesta. En otras palabras, introduce un tiempo máximo de espera del mensaje de respuesta.
Reservado
_ envelope: No se emplea, quedando reservado para llevar la información del transporte del mensaje (tiempos, rutas,...)

Con estas definiciones, el siguiente ejemplo de mensaje FIPA-ACL
(request
:sender (agent-identifier :name i)
:receiver (set (agent-identifier :name j))
:content
((action (agent-identifier :name j)
(deliver box017 (loc 12 19))))
:protocol fipa-request
:language FIPA-SL
:reply-with order567)

serviría para que un agente *i* pidiese a un agente *j* que trasladase una caja identificada como box017 a la localización (12,19), mediante un protocolo .

_parequest, con un contenido en lenguaje FIPA-SL, e indicando que las respuestas a este mensaje deben contener el valor order567 en su parámetro in-reply-to.

7. **Protocolos de interacción FIPA-ACL:** Las conversaciones entre agentes suelen seguir unos ciertos patrones en la secuencia de mensajes intercambiados. A cada una de estas secuencias típicas se les llama *protocolo de conversación.*

   El protocolo de interacción aparece en el campo protocol. Esto permite identificar el protocolo de comunicación empleado por ambas partes en la conversación actual y de este modo controlar el flujo lógico de mensajes. Así, por ejemplo, se evita que un agente responda con un cfp (acto de pedir propuestas para ejecutar una acción) ante un mensaje correspondiente a un acto de comunicación propose.

   FIPA define actualmente los siguientes 11 protocolos de interacción a saber:
   FIPA Propose
   FIPA Request
   FIPA Request When
   FIPA Query
   FIPA Contract Net
   FIPA Iterated Contract Net
   FIPA English Auction
   FIPA Dutch Auction
   FIPA Brokering
   FIPA Recruiting
   FIPA Subscribe

   Sin embargo, a modo de ejemplo, analizaremos en este punto y de manera breve el protocolo *FIPA Request*. Este protocolo permite a un agente pedir a otro agente que lleve a cabo una acción [54].

   El *iniciador* (agente que comienza el protocolo de comunicación) ejecuta la petición mediante el acto comunicativo request. El *participante* (agente que responde al acto de comunicación del iniciador) procesa el mensaje correspondiente y decide si aceptar o rechazar la petición. La respuesta tiene lugar mediante los actos comunicativos agree (opcional) y refuse respectivamente.

Una vez que el participante ha aceptado el request inicial, puede mandar los siguientes actos de comunicación al iniciador.

failure si el intento de llevar a cabo la acción ha fracasado
inform-done si ha conseguido llevar a cabo la acción y solamente quiere informar que se ha completado
inform-result si ha completado la tarea y además quiere comunicar los resultados al iniciador

3.6 ARQUITECTURAS DE MAS

Aparte de la comunicación, otros de los elementos claves en los sistemas multiagente es su arquitectura, es decir, aspectos relativos, entre otros, a su distribución, organización, gestión y estructura de comunicación. De todos los modelos existentes, elaborados por diversos grupos de investigación, en esta sección detallaremos el modelo más significativo, a nuestro juicio, de arquitectura de agentes.

3.6.1 Arquitectura de una plataforma FIPA: Gestión de Agentes.

FIPA establece una arquitectura de gestión una comunidad de agentes, imponiendo en su estándar una serie de elementos que facilitan dicha gestión. En este contexto, define la plataforma de agentes (*Agent-Platform, AP*) describiendo únicamente el comportamiento externo, dejando las decisiones de diseño a cada desarrollador. De esta forma, establece el marco en que los agentes FIPA existen y operan, así como el modelo de referencia para la creación, registro, localización, comunicación, migración y destrucción de los agentes [53, 55].

Las entidades contenidas en el modelo de referencia mostrado en la Figura 3.2 representan únicamente servicios, no implicando ninguna configuración física concreta. En este modelo, se definen los siguientes componentes lógicos: agente, facilitador de directorios, sistema gestor de agentes, servicio de transporte de mensajes y software.

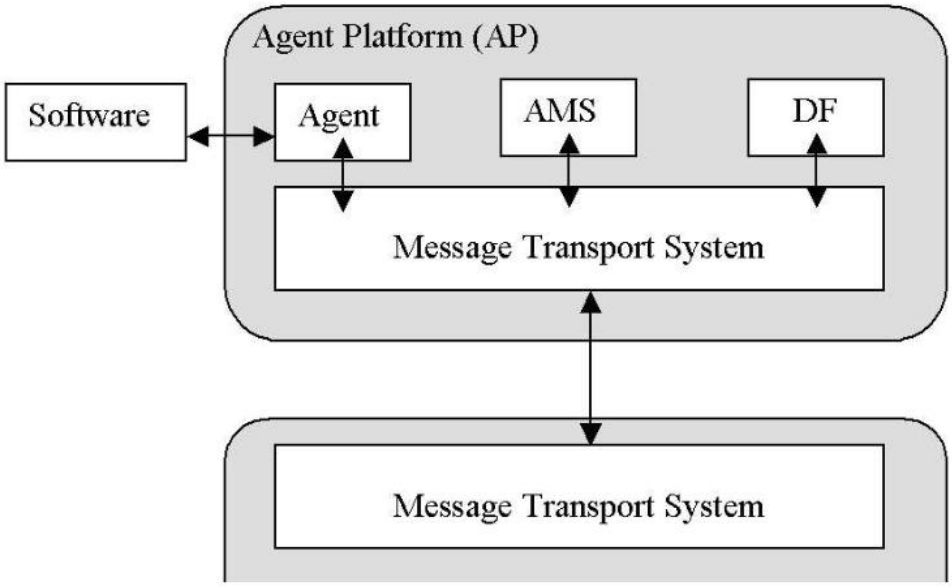


Figura 3.2.: Modelo Referencial FIPA [53]

1. **Agente:** Es el actor fundamental de la plataforma[56], que combina uno o más servicios en un modelo de ejecución integrado y unificado que podría incluir accesos a un software externo, usuarios humanos y diversas facilidades de comunicación. Un agente debe tener al menos un *propietario* (ya sea por parte de una organización al que agente está afiliado o por parte de un usuario humano) y debería prestar soporte a varias nociones de identidad. En este sentido, se define el identificador de agente (*Agent Identifier*, *AID*) como la etiqueta que permite distinguir al agente sin ambigüedad alguna dentro del universo de agentes.

2. **Facilitador de Directorios (DF):** Es un componente obligatorio en la definición FIPA de la AP. El DF (*Directory Facilitator*) [56] proporciona servicios de *páginas amarillas* (nombre de agente - servicio) para el resto de los agentes. En palabras de FIPA, es el *guardián de confianza y benigno del directorio de los agentes.* Es *de confianza* en el sentido de que se esfuerza por mantener una lista de los agentes completa, actualizada y exacta.

   Es *benigno* porque proporciona la información más reciente acerca de los agentes de forma no discriminatoria a todos los agentes autorizados. Pueden existir múltiples DFs en la misma AP. Además, los DFs pueden estar organizados según una arquitectura federativa.

Los agentes pueden registrar sus servicios con el DF o pedirle información sobre los servicios ofrecidos por otros agentes. Cada agente que desee anunciar sus servicios a otros agentes, debería en primer lugar encontrar el DF adecuado para poder solicitar su registro. Dentro del lenguaje FIPA ACL el registro lleva a cabo mediante un *register* embebido dentro de un acto de comunicación request. No obstante este registro inicial no obliga al agente a prestar el servicio indicado, pudiéndose negar a llevarlo a cabo (mediante el refuse correspondiente). El DF no garantiza la validez o exactitud de la información proporcionada por el agente que se registra, ni controla el ciclo de vida de ningún agente.

Un acto del desregistro tiene como consecuencia que el DF se libera del compromiso de facilitar información acerca de los servicios suministrados por el agente en cuestión. Del mismo modo que es posible el desregistro, un agente puede modificar la información de sus servicios en el DF en cualquier instante. Para ambas acciones, un DF debe ser capaz de procesar mensajes con las performativas *register*, *deregister*, *modify* y *search*.

El DF de la AP presenta un AID reservado correspondiente al nombre local df.

3. **Sistema Gestor de Agentes (AMS):** Es otro de los componentes obligatorios de la AP, existiendo solamente un AMS [56] (*Agent Management System*) por cada AP.

   El AMS ejerce la supervisión sobre el acceso y el uso de la AP, como la creación y destrucción de agentes, el registro de nuevos agentes, aspectos de movilidad y de control del canal de comunicación y los recursos compartidos.

   El AMS mantiene un directorio de los AIDs que contiene las direcciones de los agentes residentes en la AP en un momento dado. De este modo, el AMs ofrece un servicio de *páginas blancas* (nombre de agente - dirección), pudiendo de este modo ser consultado sobre las capacidades de los agentes de su AP. Para ello, cada agente debe registrarse en el AMS para poder obtener un AID válido. El registro en el AMS implica la autorización de acceso al MTS de la AP para poder recibir y transmitir mensajes. El AMS comprueba la validez de la descripción suministrada por el agente, en particular, la unicidad local del campo nombre del agente en el AID.

   Del mismo modo que en el caso del DF, las descripciones de los agentes en el AMS pueden ser modificados en cualquier momento, aunque se encuentra limitada por la autorización del AMS. El ciclo de vida de un agente termina con su desregistro en el AMS. Tras este desregistro, el AID del agente en cuestión puede ser eliminado del directorio, estando disponible para otros agentes que lo soliciten.

Un AMS representa la máxima autoridad en la gestión de la AP, de modo que puede solicitar a un agente que lleve a cabo una función relacionada con la gestión de la AP, por ejemplo, que abandone la plataforma (a través de la función *quit*). En caso de que su petición sea ignorada, dispone de autoridad para hacer cumplir esta petición.

Las funciones soportadas por el AMS son: *register*, *deregister*, *modify, search* y *get-description*. El siguiente código FIPA ACL, como se muestra en [53], es ilustrativo del registro de un agente en el AMS.

```
(request
:sender
(agent-identifier
:name dummy@foo.com
:addresses (sequence iiop://foo.com/acc))
:receiver (set
(agent-identifier
:name ams@foo.com
:addresses (sequence iiop://foo.com/acc))
:language FIPA-SL0
:protocol FIPA-Request
:ontology FIPA-Agent-Management
:content
(action
(agent-identifier
:name ams@foo.com
:addresses (sequence iiop://foo.com/acc))
(register
(ams-agent-description
:name

(agent-identifier
:name dummy@foo.com
:addresses (sequence iiop://foo.com/acc))
:state active))))
```

4. **Servicio de Transporte de Mensajes (MTS):** Es el método de comunicación por defecto entre los diversos agentes, ya sea entre los agentes que se encuentran dentro de la misma plataforma o con agentes residentes en otras APs. Todos los agentes FIPA deben tener acceso al menos a un MTS [56] (*Message Transport Service*) y solamente pueden ser enviados a un MTS los mensajes dirigidos a un agente. Estos servicios de transporte son proporcionados por un Canal de Comunicación de Agentes *(Agent*

Communication Channel, ACC), por lo que en la literatura se suelen emplearse indistintamente ambos términos.

Cuando un agente A desea mandar un mensaje a un agente B residente en otra AP existen tres opciones para poder hacerlo (ver Figura 3.3):

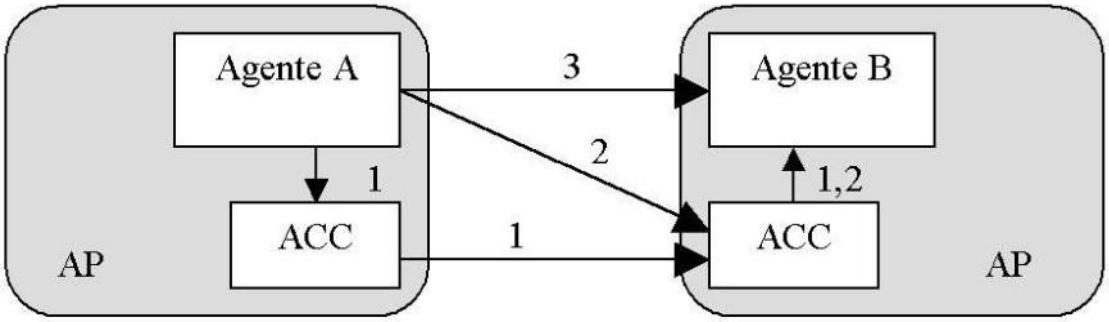


Figura 3.3.: Tres métodos de comunicación entre agentes de diferentes APs [55].

- El agente A se lo manda primero al ACC de su AP. Este ACC se encarga de entregárselo al ACC de la AP donde reside el agente B, quien se encargará, a su vez, de enviárselo al agente destinatario. En esta gestión, los ACCs pueden acceder a otros servicios de la AP (como el AMS o el DF).

- El agente A envía el mensaje directamente al ACC de la plataforma donde está registrado el agente B. Este ACC remoto es el encargado de reenviarlo al agente B.

- El agente A manda directamente el mensaje al agente B, empleando un mecanismo de comunicación directa. Son los dos agentes, A y B, los encargados de manejar la transferencia de mensajes, direccionamiento, buffer de mensajes y cualquier posible mensaje de error [55].

- El modelo de comunicación entre agentes es asíncrono. Por tanto, El ACC no se queda bloqueado ante el envío o recepción de mensajes.

- Existen colas de envío y recepción de mensajes.

- Existen políticas de gestión de colas de mensajes.

5. **Software:** Describe todos los conjuntos de instrucciones no correspondientes a ningún agente, pero que pueden ser accedidos por los mismos. Los agentes pueden acceder al software, por ejemplo, para añadir nuevos servicios, adquirir nuevos protocolos de comunicación o de seguridad... [53].

6. **Plataforma de Agentes (AP):** Proporciona la infraestructura física en que los agentes pueden ser desplegados. Consiste en la(s) máquina(s), el sistema operativo, el software de soporte de agentes, los componentes de gestión de agentes FIPA (AMS, DF y MTS) y los agentes. El diseño interno de la plataforma no se ve ligado a ninguna estandarización y depende de los desarrolladores. Obsérvese que los agentes de la AP (*Agent Platform*) pueden estar distribuidos en varias máquinas.

7. **El ciclo de vida de un agente:** De lo visto en los apartados anteriores se deduce que los agentes FIPA tiene una existencia física en una AP y emplean las facilidades ofrecidas por ella para desarrollar sus actividades. En este contexto, un agente tiene un ciclo de vida físico que debe ser gestionado por la AP.

   El ciclo de agente (ver Figura 3.8):
   - Está ligado a la AP: Un agente es físicamente gestionado en el interior de una AP y por tanto el ciclo de vida de un agente estático está siempre ligado a una AP específica.

   - Es independiente de la aplicación: El modelo de ciclo de vida de un agente es independiente del sistema de cualquier aplicación y define únicamente los estados y las transiciones del servicio del agente en su ciclo de vida.

   - Es orientado a instancias: El agente descrito en el modelo de ciclo de vida se asume que es una instancia, esto es, un agente que tiene un nombre único y es ejecutado independientemente.

   - Único: Cada agente tiene un único estado en cada instante dentro de su ciclo de vida y ligado a una única AP.

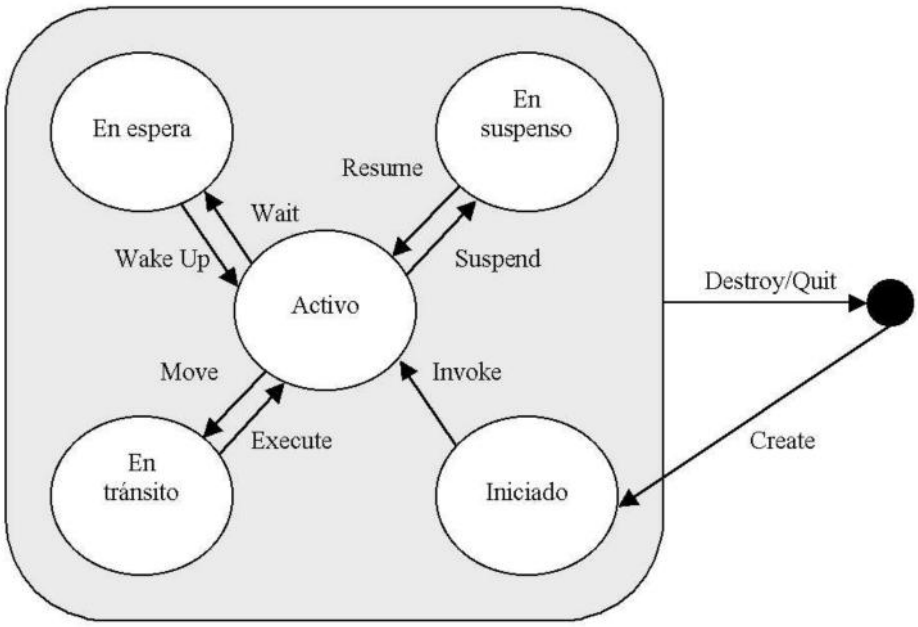


Figura 3.4.: Ciclo de vida de un agente FIPA dentro de una AP [53]

Los estados por los que puede pasar un agente en su ciclo de vida son *activo*, *iniciado*, *en espera*, *suspendido*, *en tránsito*20 y *desconocido*. Según el estado del agente, el modo de reparto de mensajes puede modificarse.

Activo: el MTS reparte los mensajes al agente de modo normal.

Iniciado/ En Espera/ Suspendido: El MTS almacena los mensajes en un buffer hasta que el agente alcanza un estado activo (o los reenvía a una nueva dirección si el sistema lo requiere)

En tránsito: El MTS almacena los mensajes en un buffer hasta que el agente alcanza un estado activo (no se ha podido completar el traslado a otra AP, o este traslado ha tenido lugar) o los reenvía a una nueva dirección si el sistema lo requiere.

Desconocido: El MTS almacena los mensajes en un buffer o los rechaza, dependiendo de la política implementada.

Las transiciones de estado de los agentes son las siguientes:

Create Creación o instalación de un nuevo agente.
Invoke Invocación de un nuevo agente.
Destroy Finalización forzada de un agente. Solamente puede ser iniciada por el AMS y no puede ser ignorada por el agente.

Quit Finalización elegante de un agente. Puede ser ignorada por el agente.
Suspend Coloca al agente en un estado de suspensión. Puede ser iniciado tanto por el AMS como por el agente.
Resume Activa al agente desde un estado de suspensión. Sólo puede ser iniciado por el AMS.
Wait Coloca al agente en un estado de espera. Solamente puede ser iniciado por el agente.
Wake Up Activa al agente desde un estado de espera. Sólo puede ser iniciado por el
AMS.
Move Solamente para agentes móviles. Coloca al agente en un estado en tránsito. Solamente puede ser iniciado por el agente.
Execute Solamente para agentes móviles. Activa al agente desde un estado en tránsito. Sólo puede ser iniciado por el AMS.

8. **Otras Especificaciones del Estándar FIPA:** En apartados anteriores se ha visto las especificaciones dadas por FIPA referidas a la comunicación y arquitectura de los sistemas multiagente. No obstante, existen otras especificaciones objeto de estandarización y con ello de compatibilidad de los MAS. Como en los anteriores casos, FIPA no especifica cómo debe ser implementado los elementos a estandarizar, sino que únicamente describe cómo debe ser su comportamiento externo.

   La labor de elaborar los posibles estándares dentro de FIPA corre a través de una serie de comités técnicos. A fecha de escritura de este trabajo, se encuentran activos siete comités encargados de los aspectos relativos a:
   - Protocolos de interacción: Trabaja en el desarrollo de una segunda generación de protocolos de interacción. Por ejemplo, ya existe un documento de propuesta de un nuevo protocolo de interacción para el mecanismo de recuento de Borda [57] de búsqueda de consenso entre varias partes.
   - Metodologías: Trabaja en la identificación de una metodología para el desarrollo de MAS.
   - Modelado: Intenta desarrollar una semántica común, meta-modelo y sintaxis para las metodologías basadas en agentes que sean independientes del fabricante.
   - Ontologías: Se encarga de los métodos de estandarización para el conocimiento compartido y filtrado a través de una representación ontológica que permita a los sistemas automatizar el procesamiento de mensajes con respecto a una clasificación semántica de referencias internas.
   - Seguridad: Su objetivo es liderar la investigación y desarrollo de la seguridad para sistemas multiagente.

- Semántica: Trabaja actualmente en la adopción de un nuevo marco semántica que dé cuenta de los actos comunicativos y protocolos de interacción FIPA, así como a un número de construcciones tales como contratos, políticas, descripciones de agentes...
- Servicios: Se encarga de proporcionar más estructura y soporte para los servicios del marco FIPA. Anima a los desarrolladores a proporcionar un metamodelo de servicios en el cual puedan expresarse los modelos de servicios disponibles actualmente (CORBA, servicios Web, DAML-S).

El trabajo de estos comités ha dado lugar a una serie de documentos e informes técnicos que reflejan los estándares FIPA que deben ser tenidos en cuenta en el desarrollo de una aplicación que quiera ser compatible con otras aplicaciones FIPA. Las especificaciones aprobadas a fecha de escritura de este trabajo se refieren a [55]:

- Especificaciones relativas a aplicaciones (aplicación nómada, integración de agente software, asistente personal de viajes, difusión y entretenimiento audiovisual, provisión y gestión de red, asistente personal, buffer de mensajes, calidad de servicio)
- Especificaciones sobre la arquitectura abstracta (políticas y dominios).
- Especificaciones sobre la comunicación de agentes (estructura de mensajes
- ACL, servicio de ontologías, protocolos de interacción, actos de comunicación, lenguajes de contenido de mensajes).
- Especificaciones sobre la gestión de agentes (AMS, DF...)
- Especificaciones sobre el transporte de mensajes (interoperabilidad de mensajes, representaciones del mensaje como cadena de caracteres, en bits ó XML, protocolo de transporte...)

Como se puede apreciar, FIPA intenta cubrir la mayor parte posible de los aspectos de un sistema multiagente, buscando que un nuevo MAS interopere adecuadamente con otro MAS existente.

4 MODELO PROPUESTO

Con los elementos expuestos en los capítulos anteriores, el uso de ontologías permitirá realizar un mapeo de la red y presentarlo de forma abstracta y que sea comprensible por el uso de los computadores, esto facilitará el desarrollo del Modelo propuesto, el cuál inicia con el Modelo Conceptual pasando por el Modelo Ontológico, y terminando con el Modelo de Agentes.

4.1 MODELO CONCEPTUAL

En la gestión de red la mayoría de los conflictos se encuentran en dos capas a saber: la capa de acceso al medio y la capa de aplicación vistos desde el modelo TCP/IP [30]. En la primera se tienen los medios físicos y los enlaces que interconectan los equipos y dispositivos, en ellos se encuentra la información del desarrollo topológico de la red, capacidades de canales de comunicación e integración a nivel de bit. Todas las capas superiores dependen directamente de la capa de acceso al medio, si esta falla, las superiores no pueden funcionar.

La segunda se encuentra todas las aplicaciones que se utilizan para la transferencia de todos los datos en información dentro de una red de computadores, con los protocolos que soportan servicios de conexión remota y la transferencia de archivos, en esta se encuentra el dolor de cabeza de los administradores de red, esta capa funciona a través de protocolos de red, y permiten que servidores, a través de servicios, compartan, almacenen, intercambien información y datos.

Para empezar el análisis del modelo partiremos de delimitar los equipos o dispositivos que componen una red de datos, a saber:

- Equipos de cómputo
- Dispositivos de Red
- Servidores
- Equipos de enrutamiento y firewall

Para concretar el modelo primero se definirán cada uno de los términos, y sus propiedades por capa:

Equipos de cómputo [62]: es una máquina electrónica que recibe y procesa datos para convertirlos en información conveniente y útil. Un equipo de cómputo está formado, físicamente, por numerosos circuitos integrados y otros muchos componentes de apoyo, extensión y accesorios, que en conjunto pueden ejecutar tareas diversas con suma rapidez y bajo el control de un programa.

Dos partes esenciales la constituyen, el hardware, que es su composición física (circuitos electrónicos, cables, gabinete, teclado, etcétera) y su software, siendo ésta la parte intangible (programas, datos, información, etcétera). Una no funciona sin la otra.

Dependiendo de su uso y tecnología se puede encontrar los siguientes:

- Computador personal de escritorio
- Computador personal portátil
- Teléfono celular Inteligente
- Tablet.

Adicionalmente se pueden encontrar otros equipos que aunque no son unidades de almacenamiento de información son equipos que se interconectan a la red :

- Impresora
- Teléfono IP
- Scanner

Propiedades para la capa de acceso al medio:

- Nombre
- Dirección MAC
- Velocidad de Conexión

Propiedades para la capa de aplicación

- Dirección IP
- Estado de Conexión de para cada protocolo

Dispositivos de Red [63]: Es el hardware que permite comunicarse entre sí a los equipos de cómputo que hay en una red. Dependiendo del medio de comunicación se encuentra los siguientes dispositivos:

- Conmutador (switch) [63] es el dispositivo digital lógico de interconexión de equipos que opera en la capa de enlace de datos del modelo OSI. Su función es interconectar dos o más segmentos de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red y eliminando la conexión una vez finalizada esta.

- Punto de acceso inalámbrico [63] (en inglés: Wireless Access Point, conocido por las siglas WAP o AP), en una red de computadoras, es un dispositivo de red que interconecta dispositivos móviles a través de radio frecuencia.

Servidor [64]: Un servidor es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente (Equipo de cómputo) y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, incluso en computadoras dedicadas a las cuales se les conoce individualmente como "el servidor". En la mayoría de los casos una misma computadora puede proveer múltiples servicios y tener varios servidores en funcionamiento. La ventaja de montar un servidor en computadoras dedicadas es la seguridad.
Aunque se pueden tener extensa cantidad de servidores (más de 65.000 puertos TCP disponibles), en nuestro caso se mencionarán los de mayor uso en las redes de computadores.

Los servidores de mayor uso en una red de datos son los siguientes:

- Servidor DHCP: servidor encargado del suministro dinámico de direcciones IP a los equipos.
- Servidor HTTP: Servidor que provee el servicio de acceso a página Web corporativa a los equipos conectados en la red.
- Servidor DNS:
- Servidor de gestión
- Planta telefónica.
- Servidor de Bases de datos

Propiedades para la capa de acceso al medio:

- Nombre
- Dirección MAC
- Velocidad de Conexión

Propiedades para la capa de aplicación

- Dirección IP
- Disponibilidad de servicio.
- Listas de control de acceso.

Equipos de enrutamiento y firewall: En este aparte se encuentras dos dispositivos

Firewall[65]: Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Si el tráfico entrante o saliente cumple con una serie de Reglas especificadas, entonces el tráfico podrá acceder o salir de la red o equipo de cómputo sin

restricción alguna. En caso de no cumplir las reglas el tráfico entrante o saliente será bloqueado.

Propiedades para la capa de acceso al medio:
- Nombre
- Dirección MAC
- Velocidad de Conexión

Propiedades para la capa de aplicación
- Dirección IP
- Reglas de acceso a protocolos
- Listas de control de acceso.

Enrutador (Router) [66]: Un router es un dispositivo de red que permite el enrutamiento de paquetes entre redes independientes. Este enrutamiento se realiza de acuerdo a un conjunto de reglas que forman la tabla de enrutamiento. Es un dispositivo que opera en la capa 3 del modelo OSI o capa 2 del Modelo TCP/IP.

Propiedades para la capa de acceso al medio:
- Nombre
- Dirección MAC
- Velocidad de Conexión

Propiedades para la capa de aplicación
- Dirección IP
- Listas de control de acceso.
- Conexión a red externa

Con las definiciones de las propiedades y las entidades que participan del modelo, se generan los grafos que muestran las relaciones para realizar la gestión en una red de datos.

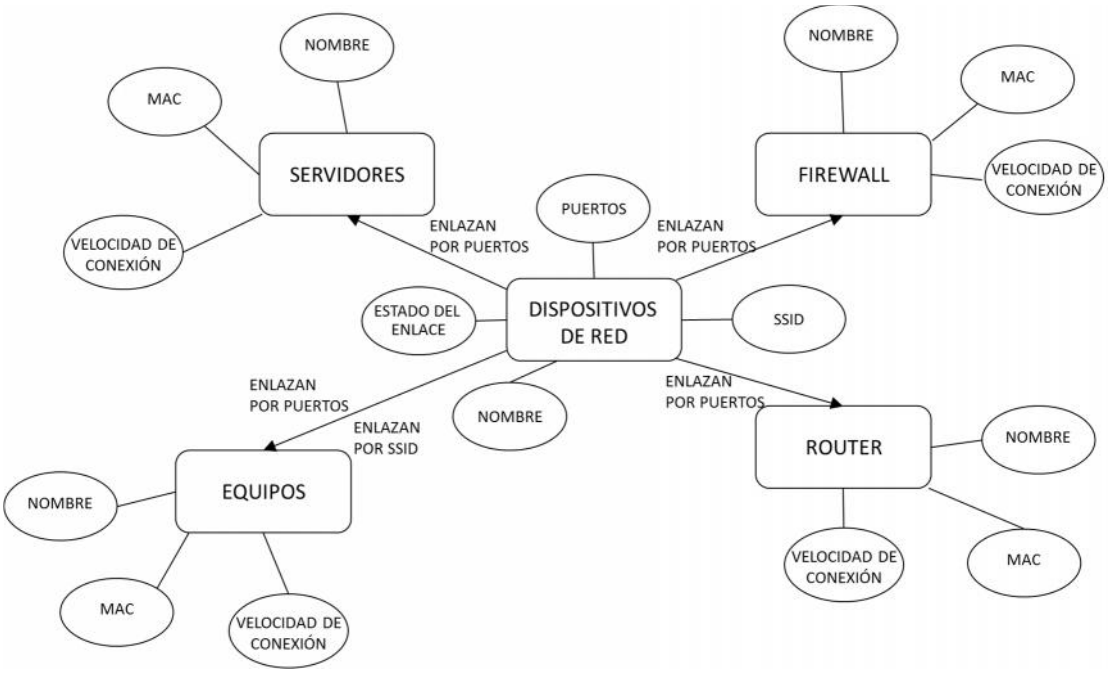


Figura 4.1 Modelo conceptual capa de acceso al medio

Para la definición del grafo a nivel de aplicación, figura 4.2, el análisis que se hace es en referencia a los protocolos y servicios que se encuentran en la capa de aplicación, estos son los que permiten a los usuarios la transferencia de información, la conexiones remotas y la integración de aplicaciones, es en este punto donde el elemento de gestión principal es el Firewall (Muro de fuego) el cual es el encargado del control de todas las comunicaciones que viajan a través de la red, este define las políticas que facilitan el intercambio de información.

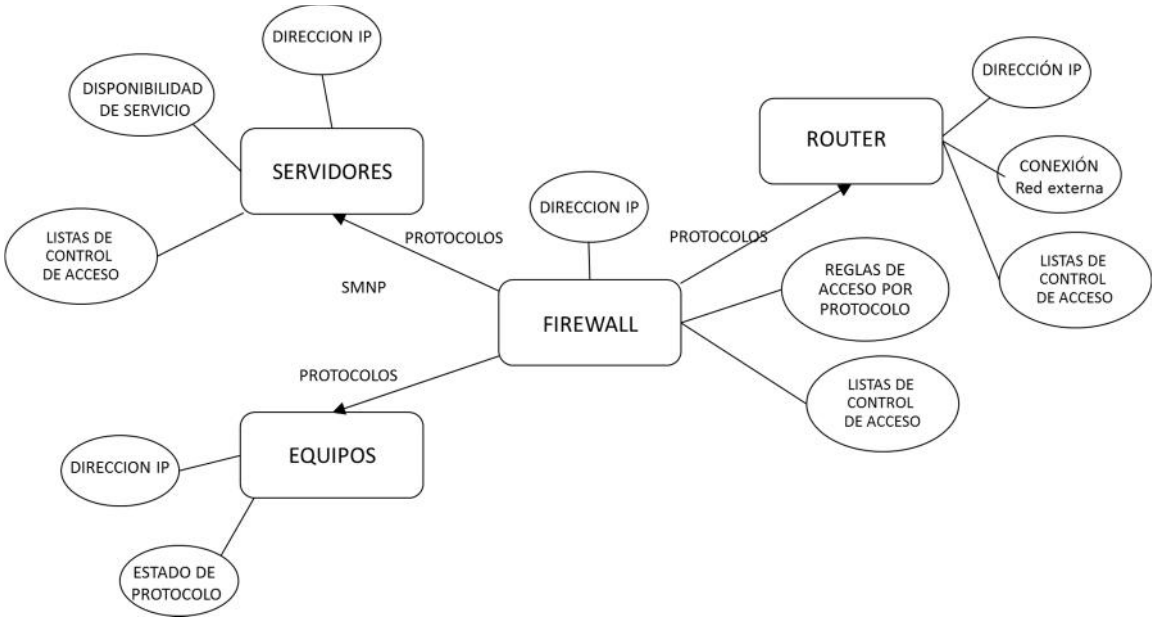


Figura 4.2 Modelo conceptual capa de aplicación

Así encontramos que dependiendo de la capa del modelo TCP-IP, se encontrarán diversos elementos encargados del control de la información, en la capa 2 nivel de enlace, se encuantra los dispositivos de red (switch, puntos de acceso, etc) como los encargados del control de la tranferencia de bits, a través de la conexión física de los equipos. En la capa de aplicación encontramos al firewall como eje central de la transferencia de datos, y este controla las políticas de conexión.

Como se mostró en el apartado anterior se definieron los equipos y dispositivos que conforman el modelo, ahora es necesario incluir los mecanismos de gestión para así tener una red de datos completa incluido su sistema integrado de gestión. En las siguientes figuras se muestra el modelo de la capa de enlace con su sistema de gestión, figura 4.3, además de en la figura 4.4 el modelo de la capa de aplicación en la cual se incluye incluye los mecanismos de gestión que se utilizan:

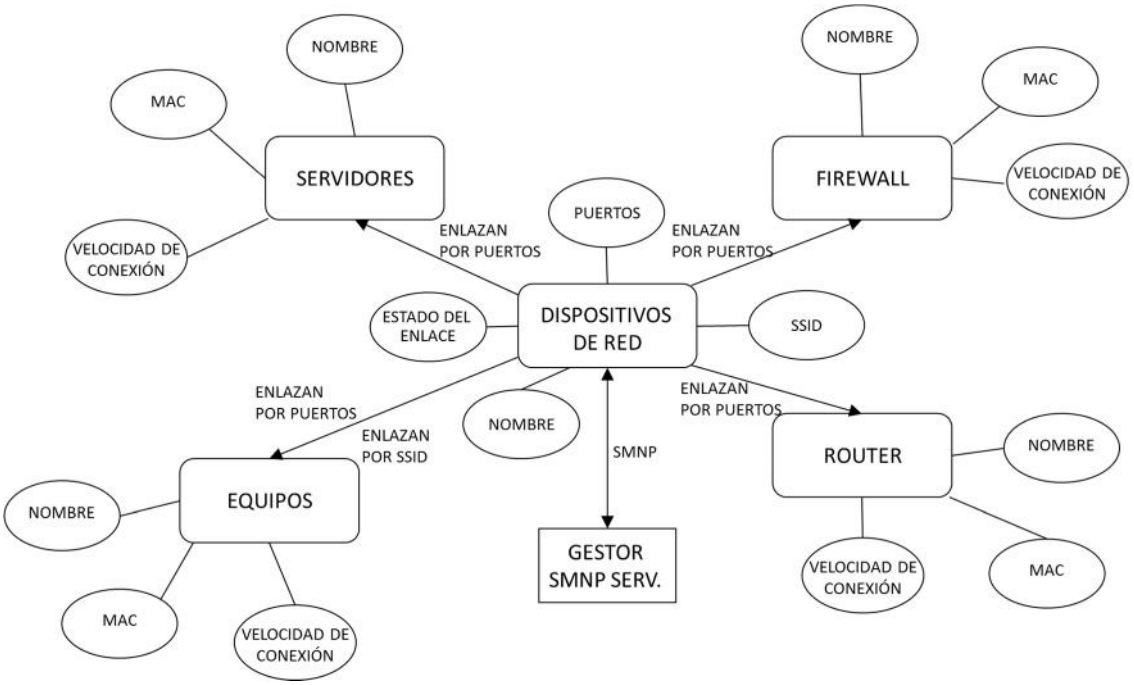


Figura 4.3 Modelo conceptual capa de acceso al medio con mecanismo de gestión

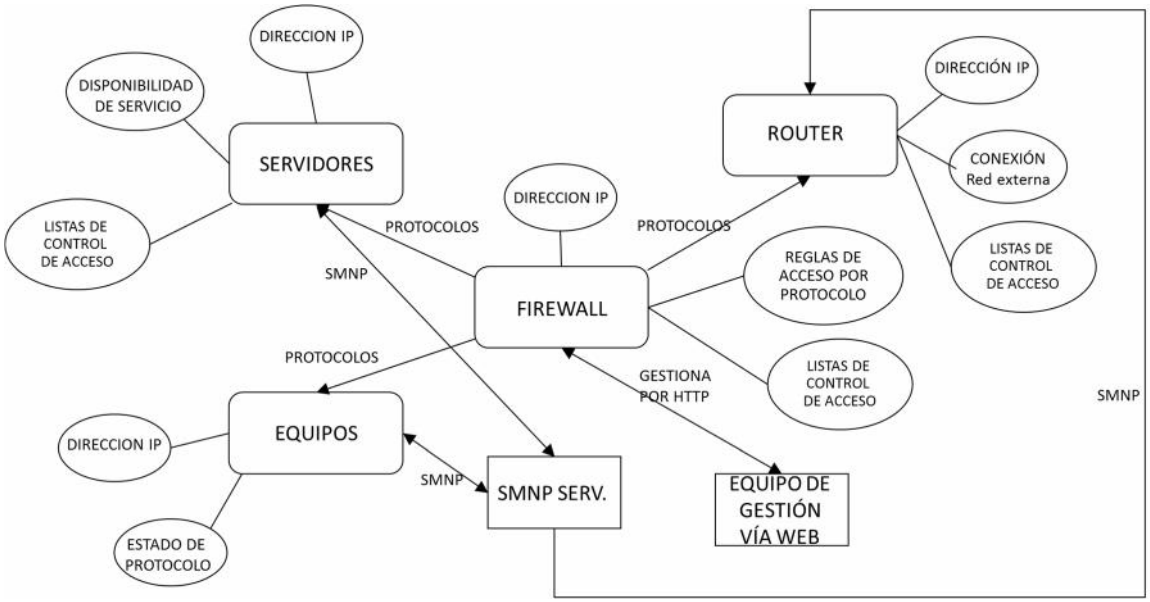


Figura 4.4 Modelo conceptual capa de aplicación con mecanismo de gestión

Obteniendo como resultado varios modelos de gestión inmersos dentro de la gestión de red, dificultando la labor del administrador de red, y como resultado negativo, que este tenga gastar mayores tiempos en manejo y control de una red.

4.2 MODELO ONTOLÓGICO

La utilización de un lenguaje de ontologías para definir la información de gestión ofrece ventajas adicionales, la primera de las cuales es la posibilidad de utilizar herramientas existentes para trabajar y razonar con las ontologías (por ejemplo, motores de inferencia utilizados en inteligencia artificial).

Otra ventaja, comentada en el apartado anterior, es que la ontología de gestión puede llegar a incluir la definición de reglas de comportamiento de la información de gestión. De esta forma, las definiciones de comportamiento que se suelen incluir implícitamente en las definiciones de la información de gestión (se indican en lenguaje natural, o se suponen), pueden llegar a ser expresadas de forma integrada con las definiciones de información de gestión, y en su mismo lenguaje (lenguaje de ontologías).

Es decir: esta aproximación supone que las definiciones de comportamiento quedan ahora expresadas formalmente en el mismo lenguaje de información de gestión, y pueden ser interpretadas y validadas por un gestor semántico (basado en ontologías) para trabajar y razonar sobre ellas. Todas las definiciones de gestión, tanto las de la información de gestión (MIBs) como las reglas de comportamiento, se integran ahora en una misma ontología de gestión de red.

Como lenguaje de definición de ontologías, se propone el uso de OWL, el cual es un lenguaje de ontologías de propósito general definido para la Web Semántica que contiene todas las construcciones necesarias para describir formalmente la mayor parte de las definiciones de información de gestión: clases y propiedades, con jerarquías, y restricciones de rango y de dominio. SWRL[66] extiende el conjunto de axiomas de OWL para incluir reglas condicionales.

Basados en modelo conceptual y unificando las dos capas en un mismo nivel se desarrolla el modelo de la ontología que nos servirá para el desarrollo de la ontología en el aplicativo, así en la figura 4.5 se puede ver la integración de las dos capas, y el modelo.

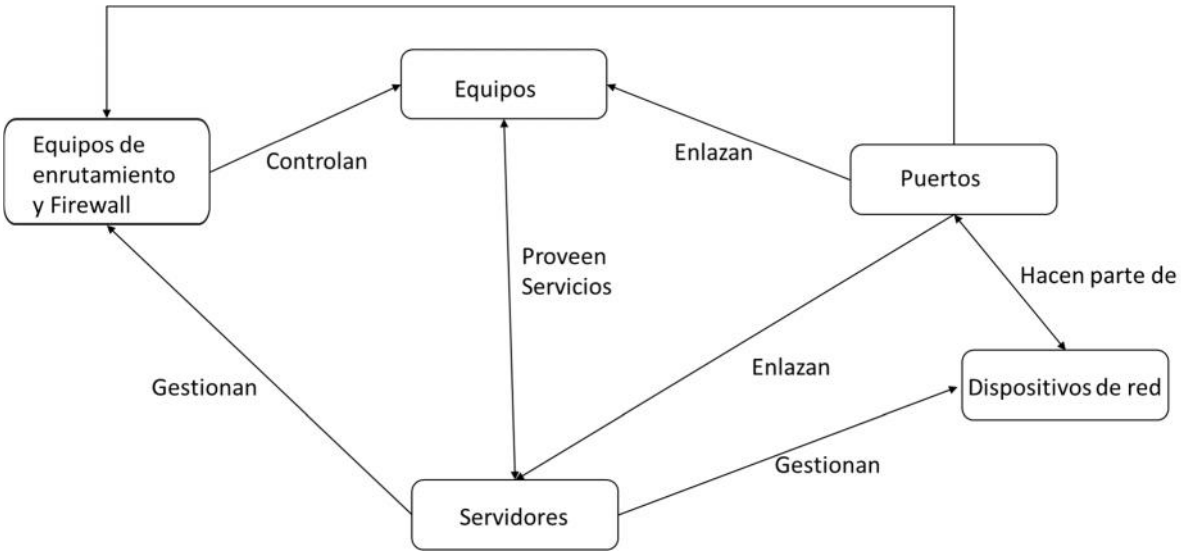


Figura 4.5 Modelo de ontológico

En este modelo aparece una nueva entidad los puertos, que surgen como necesidad que permita la interconexión de todos equipos y dispositivos, para lo cual procederemos a definir su terminología.

Puertos [67]: Los medios de transmisión son las vías por las cuales se comunican los datos. Dependiendo de la forma de conducir la señal a través del medio o soporte físico, se pueden clasificar en dos grandes grupos:

- medios de transmisión guiados o alámbricos.
- medios de transmisión no guiados o inalámbricos.

En ambos casos las tecnologías actuales de transmisión usan ondas electromagnéticas. En el caso de los medios guiados estas ondas se conducen a través de cables o “alambres”. En los medios inalámbricos, se utiliza el aire como medio de transmisión, a través de radiofrecuencias, microondas y luz (infrarrojos, láser).

Para finalizar desarrollando en modelo de clases y subclases de la ontología que se ilustra en la figura 4.6, en la cual se detallan con diferentes colores las relaciones entre las diferentes clases de la ontología, para los elementos para gestión de equipos por SMNP verde, rojo para el control de enlace de los dispositivos de red, azul para los servicios que se proveen desde los servidores hasta los equipos, y con café el control por políticas y reglas que hace el firewall hacia los equipos.

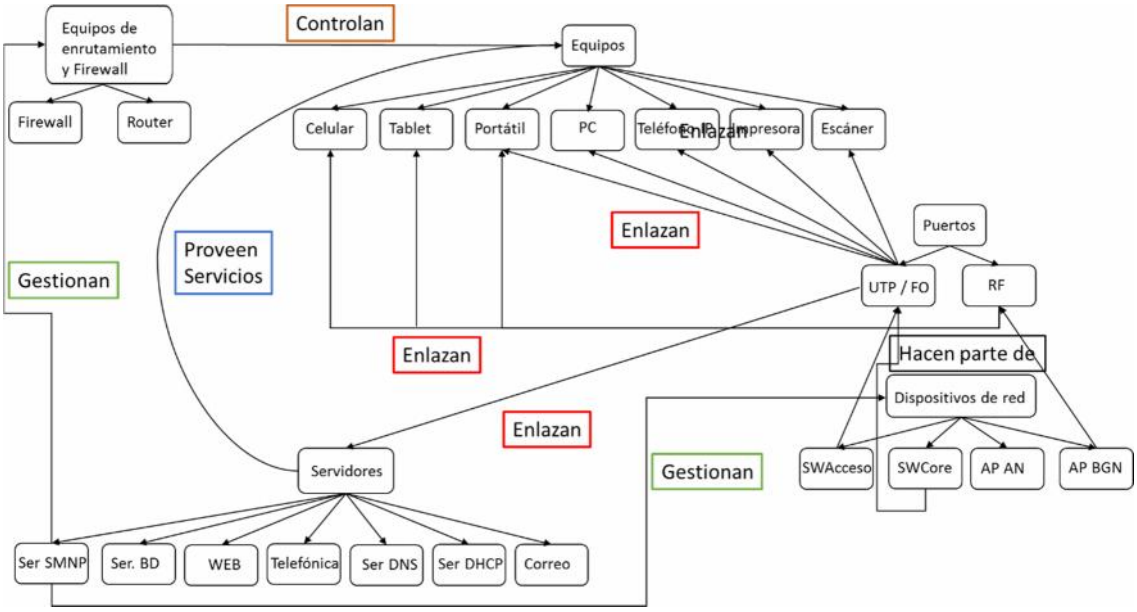


Figura 4.6 Modelo ontológico clases y subclases

Para implementar la ontología se utilizará el programa Protégé, en la figura 4.7 observamos la ontología realizada.

Protégé [70] es una herramienta para el desarrollo de Ontologías y Sistemas basados en el conocimiento creada en la Universidad de Stanford, desarrollada en JAVA y puede funcionar perfectamente bajo diferentes sistemas operativos.

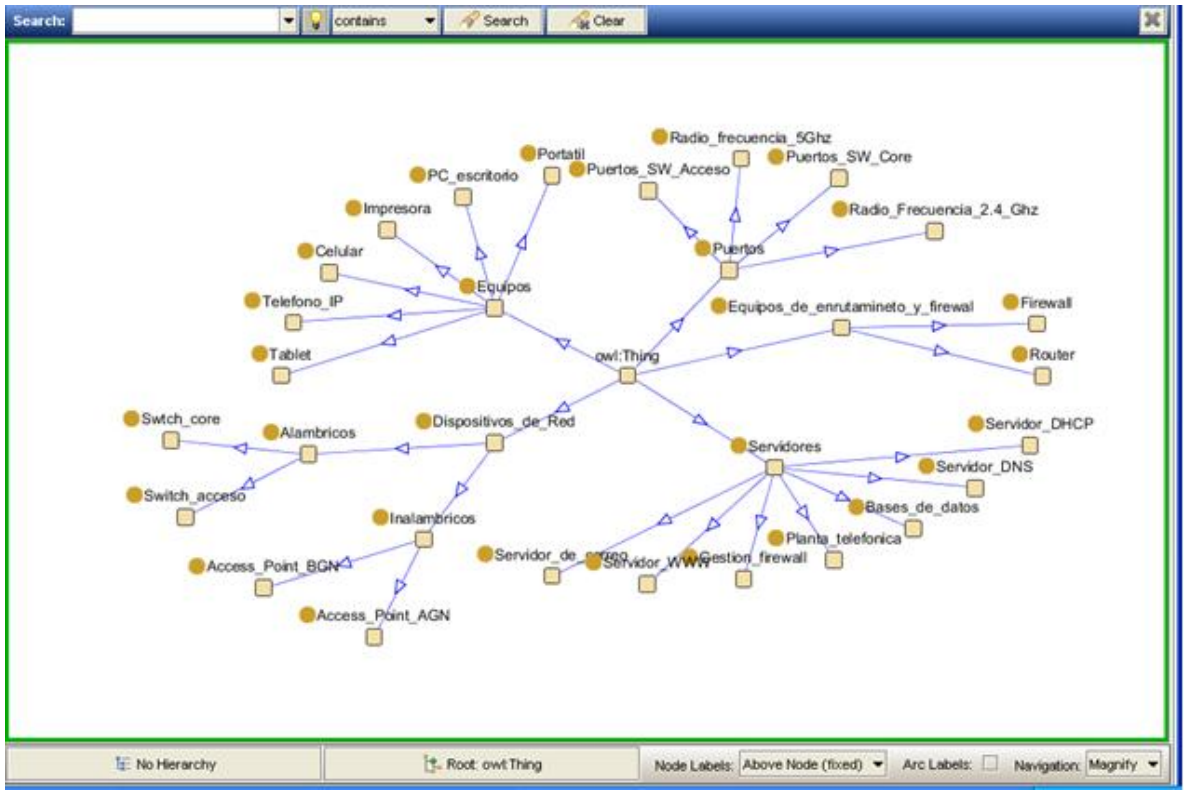


Figura 4.7 Ontología implementada en protégé.

Con las clases y las subclases definidas, se crean las propiedades de los objetos y los datatypes, con lo cual se le da capacidad semántica a las clases, las tablas 4.1 y 4.2 las detallan:

Tabla 4,1 propiedades de los objetos de la Ontología

Propiedad Objetc	**Dominio**		**Rango**			
Acceso a servidor Bases de Datos	Bases de Datos		Equipos			
Acceso a servidor Web	Servidor WWW		Equipos			
Acceso a Red Externa	Router		Equipos			
Acceso a Servidor de Correo	Servidor Correo		Equipos			
Conecta Equipos	Puertos SW acceso		Telefono IP	Impresora	PC escritorio	Portatil

Conecta AP BGN	Conecta Equipos BGN		Celular	Tablet	portatil	
Conecta Datacenter	Puertos SW Core		Servidores	Firewall	Router	
Entrega No extensión a Tel IP	Planta telefónica		Teléfonos IP			
Gestión Firewall	Servidor de Gestión		Firewall			
Entrega dirección IP	Servidor DHCP		Equipos			
Gestiona por SMNP	Servidor SMNP		Servidores	Router	Dispositivos de red	
Permite Uso DNS	Firewall		Equipos			
Permite Email	Firewall		Equipos			
Permite DHCP	Firewall		Equipos			
Permite FTP	Firewall		Equipos			
Permite SMNP	Firewall		Servidores	Dipositivos de red	Router	
Permite bases de datos	Firewall		Equipos			
Permite HTTP	Firewall		Equipos			
Pertenece a	Puertos SW Core	Puertos SW acceso	Alámbricos			
Puertos Inalámbricos 2.4 Ghz	Radio Frecuencia 2.4 Ghz		Access Point BGN			
Puertos Inalámbricos 5 Ghz	Radio Frecuencia 5 Ghz		Access Point AGN			
Servidor DNS para	Servidor DNS		Equipos			

Tabla 4,2 propiedades de los objetos de la Ontología

Propiedad Datatype	**Rango**				**Tipo**
Dirección IP	Equipos	Equipos de enrutamiento y Firewall	Servidores		String
Estado Conexión DNS	Equipos				Boolean
Esado Conexión SMNP	Dispositivos de red	Servidores	Router		Boolean
Estado Conexión WWW	Equipos				Boolean
Estado Enlace	Puertos				Boolean
Estado Conexión Correo E	Equipos				Boolean
Frecuencia WIFI	Celular	Tablet	portátil		String
MAC	Equipos	Equipos de enrutamiento y Firewall	Servidores		String
Nombre	Equipos	Equipos de enrutamiento y Firewall	Servidores	Dispositivos de red	String
Numero de puertos	Switch Acceso	Switch core			Int
Reinicio servidor DNS	Servidor DNS				Boolean
Reinicio Servidor BD	Servidor bases de datos				Boolean
Reinicio Planta telefónica	Planta telefónica				Boolean
Reinicio Servidor DCHP	Servidor DHCP				Boolean
Reinicio Servidor WWW	Servidor WWW				Boolean
Reinicio Puerto	Puertos				Boolean

Velocidad de conexión	Equipos				String

Con las propiedades y las clases desarrolladas la ontología se revisa la forma en que las clases quedaron relacionados, para verificar con los modelos propuestos, y se observa en la figura 4.8

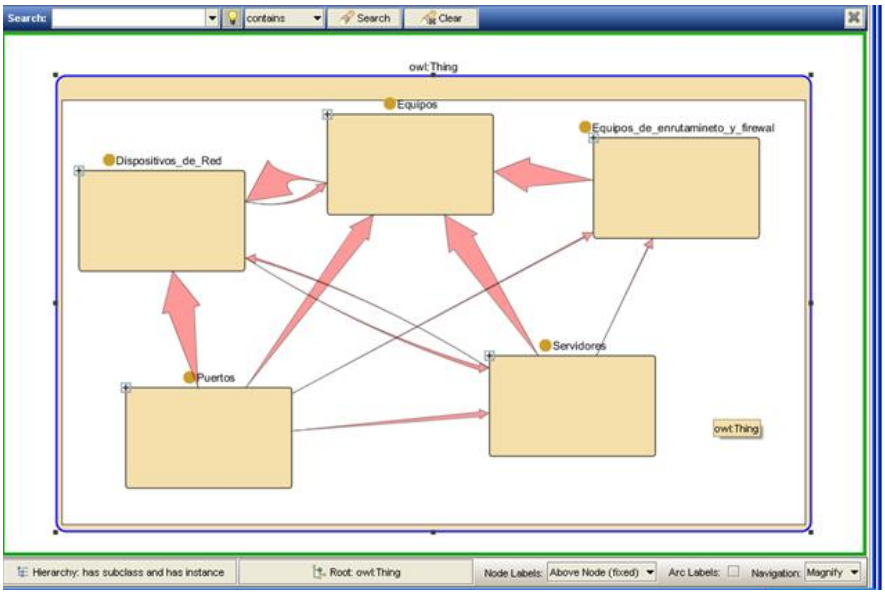
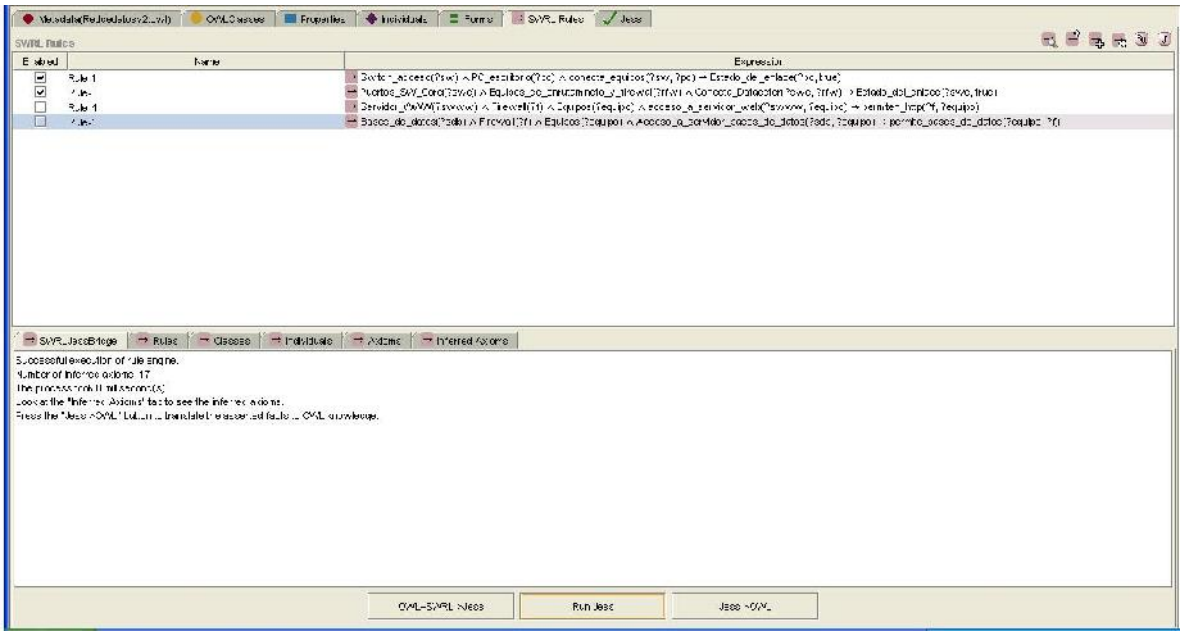


Figura 4.8 Relaciones de las clases

4.2.1 Reglas de restricción.

Una ontología especifica un conjunto de restricciones que declaran lo que debería cumplirse necesariamente en cualquier mundo posible.

En la otología para gestión de red se pueden encontrar diversas reglas de restricción, los cuales generan las inerencias que muestran las realciones del mundo real con la ontología, en la figura 4.9 se pueden detallar algunas de ellas, las cuales permiten ver en el caso ejemplo las ineferncias para determinar la cantidad de equipos y equipos de enrutamiento que están activos sobre los dispositivos de red.



Las reglas de restricción sobre un modelo de información de gestión son restricciones sobre los valores que pueden tomar las variables de gestión, en función de las condiciones del entorno.

La reglas que tenemos son:

- Switch_acceso(?sw) ∧ PC_escritorio(?pc) ∧ conecta_equipos(?sw, ?pc) → Estado_del_enlace(?pc, true)
- Puertos_SW_Core(?swc) ∧ Equipos_de_enrutamineto_y_firewal(?rfw) ∧ Conecta_Dataceter(?swc, ?rfw) → Estado_del_enlace(?swc, true)

Estas primeras reglas muestran la información sobre el estado de conexión o desconexión de un equipo sobre la red, mostrando si un equipo de un usuario en particular se encuentra encendido o apagado, de esta forma realizar los correctivos por fallas en el nivel fisco.

En las siguientes reglas se tienen las políticas de los equipos y el firewall que se asocian a las posibilidades de conexión o no de los diversos protocolos sobre los equipos, en el primer caso tenemos la posibilidad que tienen los diferentes equipos en conectarse al servidor de WWW, de forma que el primer filtro es el firewall y por último las restricciones propias del equipo, así tener de primera mano la unión de restricciones y tener un mejor control del modelo, así mismo aplicamos a los otros casos como son en servicio de bases de datos y de correo electrónico.

Para estas reglas tenemos:
Servidor_WWW(?swww) ∧ Firewall(?f) ∧ Equipos(?equipo) ∧ acceso_a_servidor_web(?swww, ?equipo) → permiten_http(?f, ?equipo)

Bases_de_datos(?sdb) ∧ Firewall(?f) ∧ Equipos(?equipo) ∧ Acceso_a_servidor_bases_de_datos(?sdb, ?equipo) → permite_bases_de_datos(?equipo, ?f)

Svr_Correo(?sem) ∧ Firewall(?f) ∧ Equipos(?equipo) ∧ Acceso_a_correo-e(?sem, ?equipo) → permite_correo-e(?equipo, ?f)

La utilización de un lenguaje de ontologías aporta ventajas adicionales las ontologías de gestión de red pueden ser utilizadas para la definición de reglas que gobiernen el comportamiento de los elementos gestionados. De esta forma, el comportamiento esperado de estos elementos es definido en forma de políticas o restricciones, pudiendo ser expresado de una manera formal mediante ontologías unidas con las definiciones de gestión, de forma que un gestor pueda trabajar y razonar con ellas.

4.3 MODELO DE GESTIÓN A TRAVÉS DE AGENTES

La modelo global propuesta se basa en un gestor que trabaja y razona con un único modelo de información de gestión representado mediante ontologías. Este gestor maneja elementos de diferentes dominios (Equipos, dispositivos de red, servidores.) desde un punto de vista común y neutral a todos ellos. La Fig. 4.12 muestra la arquitectura propuesta para el gestor con el uso de los agente.

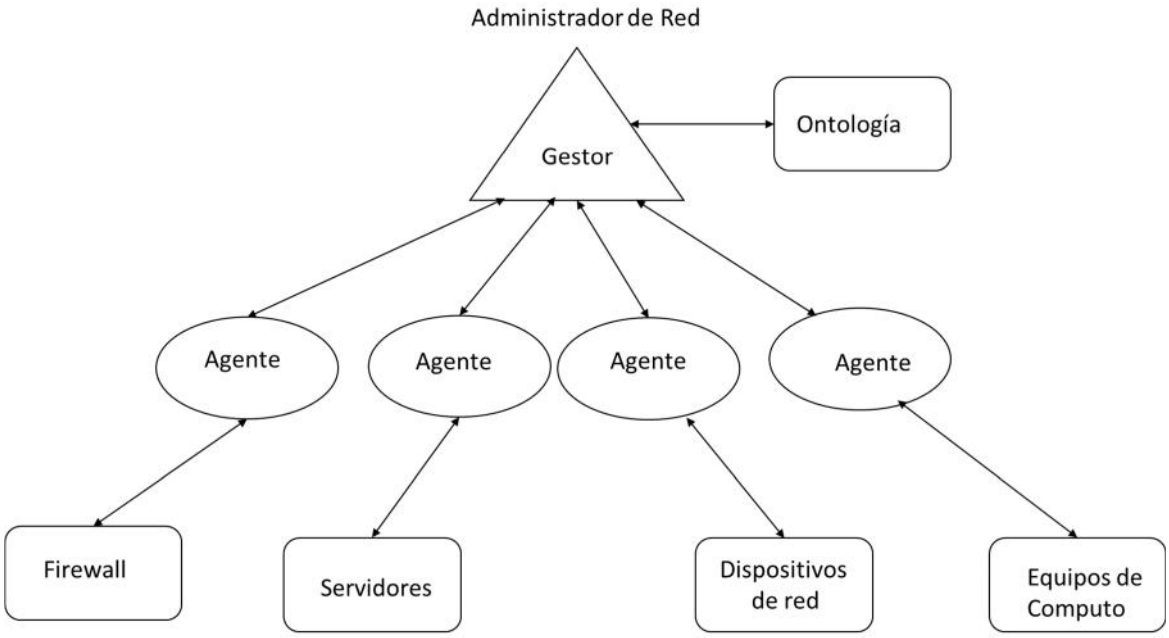


Figura 4.10 Modelo de agentes

El objetivo principal es que el gestor utilice un único modelo de información, pero en muchos casos los distintos recursos de red están definidos en distintos dominios de gestión y hay que acceder a ellos utilizando los protocolos definidos para dicho dominio. Por lo tanto es necesario traducir e integrar estas definiciones en una definición unificada, teniendo en cuenta la semántica de dichas definiciones. Es decir, un mismo recurso, definido dos veces en dos modelos diferentes, debe quedar con una única representación en el modelo unificado, y con dos traducciones a los modelos origen. Se trata por tanto no sólo de una tarea de traducir definiciones sintácticamente, sino también de integrarlas desde un punto de vista semántico.

Ahora se define cual debe ser el rol y la función de cada agente, para lo cual tenemos:

4.3.1 Agentes de monitoreo y control de dispositivos

El Agente de monitoreo y control de dispositivos recibe información de los dispositivos sobre cambios de estado, y puede cambiar la configuración de un dispositivo. Se tienen dos tipos de agente:

A. **Agentes de monitoreo y control de Switch:** Monitorea y controla los puertos de lo switch.

Funciones:

- Monitorear cada puerto del switch para determinar cambios de estado.
- Reporta el cambio de estado de estado al agente Gestor.
- Configura los puertos de switch a petición del agente gestor
- Reinicia los puertos del switch a petición del agente gestor
- Recibe de los agentes de monitoreo de equipos información de solicitud de apagado de equipos

B. **Agente de monitoreo y control de Access Point:** Monitorea los enlaces de los dispositivos inalámbricos conectados al Access point

Funciones:

- Monitorear SSID para reportar los equipos conectados al Agente gestor.
- Desconecta equipos del SSID. a petición del agente gestor
- Reinicia el Access Point a petición del agente gestor
- Recibe de los agentes de monitoreo de equipos información de solicitud de apagado de equipos

4.3.2 Agentes de monitoreo de equipos

El agente de monitoreo de equipos, es el encargado de observar que protocolos está utilizando un equipo, con el fin de reportar si este tienen permitido el uso en la red, además de detectar si un protocolo está teniendo fallas de conectividad y reportarlo al agente gestor.

Funciones

- Detectar uso de protocolos TCP.
- Reportar el uso de protocolos al Agente gestor a fin de determinar permisos de acceso a la red.
- Informar de fallas en el uso de protocolos al agente gestor.

4.3.3 Agentes de Monitoreo y control de servidores.

El agente de monitoreo de servidores es el encargado de la verificación del servicio, y de reportar la lista de control de acceso de los servidores.

Funciones:

- Monitorea el estado del servicio de los servidores.
- Reporta fallas al agente gestor
- Reiniciar servicio a petición del agente gestor
- Reportar lista de control de acceso al agente Gestor

4.3.4 Agente de monitoreo y control de Router.

El agente de monitoreo y control de router, es el agente encargado de verificar el estado del canal de acceso a red externa, y de informar la lista de control de acceso de los equipos.

Funciones

- Monitorea el estado del canal de acceso a red externa en el router
- Reportar la caída del canal al agente gestor.
- Reiniciar los puertos de salida del router a petición del agente gestor.
- Reportar lista de control de acceso al agente Gestor

4.3.5 Agente de monitoreo de Firewall.

El agente encargado de reportar el estado de las políticas de restricciones del firewall.

Funciones:

- Reportar al agente gestor el estado de las políticas de restricciones del firewall.

4.3.6 Agente Gestor

Es el agente que contiene las bases de datos de gestión de la información de la red, controla y supervisa la gestión de la red.

Funciones:

- Recibir información de los agentes de monitoreo
- Mantiene la bases de datos de listas de acceso servidores y router
- Mantienen la bases de datos de las restricciones de firewall
- Solicita a los agentes de monitoreo y control de switch el reinicio de puertos
- Solicita a los agentes de monitoreo y control de Acces point el desconectar un equipo
- Solicita a los agentes de monitoreo y control de servidores el reinicio de servicio.

En la tabla 4,3 se presenta el desarrollo de algunos de los agentes ya detallados

Tabla 4,3 agende de monitoreo y control de Switch

Agente	Monitoreo y control de Switch
Tipo	Agente Software
Papel	Monitoreo ejecutor
Capaciddes de Razonamiento y experiencia	Capacidad de negociación con otros agentes. Protocolo SNMP.
Descripción	Monitorea y controla los puertos de switch.
Objetivo	Monitorear y controlar el estado de los puertos del switch
Comunicación	Se comunica con Switch
Coordinación	Interactúa con el gestor principal, interactua a trvés de SMNP con los switch
Tarea	Mantener los enlaces del Switch
Objetivo	Monitorear, controlar y modificar el estado de los puertos del switch
Entrada	Trap SNMP
Salida	Informe de estado de puertos
Precondición	Cambios de estado puertos
Frecuencia	Permanente

Con los datos del agente se tiene el cuadro de control, ver figura 4.10

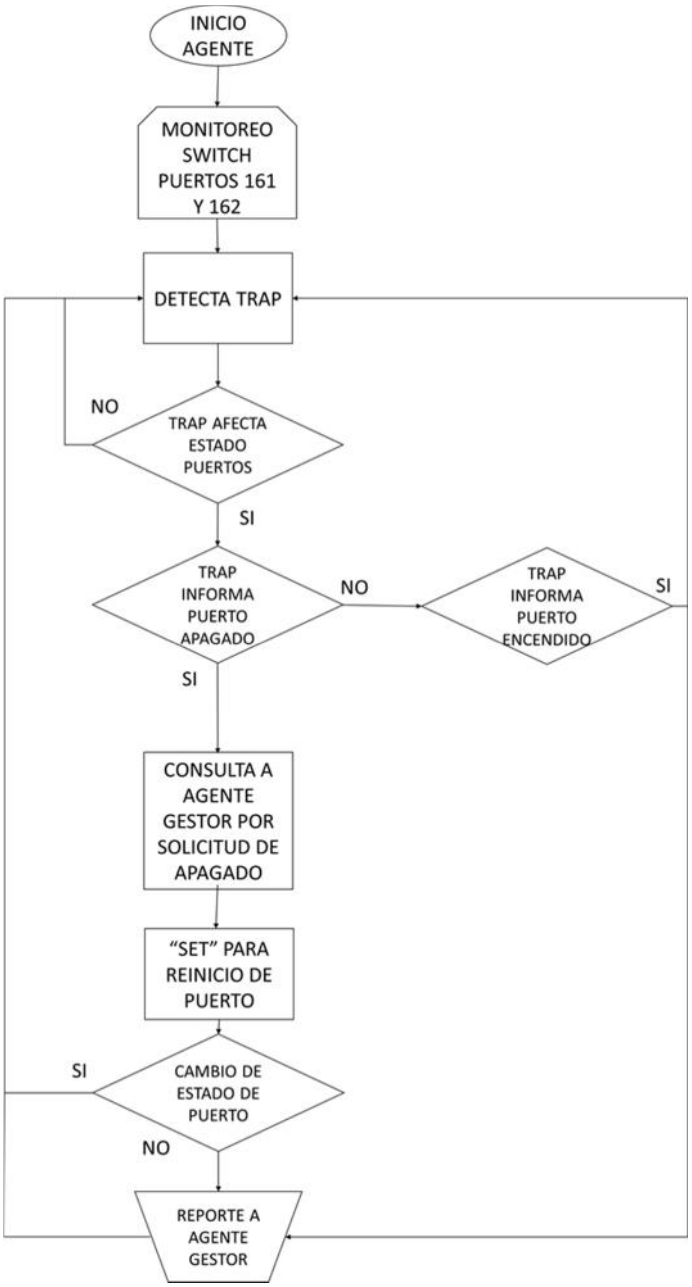


Figura 4.10 Modelo Agente monitoreo y control switch

Este capítulo desarrolla en tema de los modelos, y facilita el desarrollo de cada uno de los procedimientos, de implementación del sistema Multiagente, además de la posibilidad de especificar el comportamiento de gestión de manera formal en un lenguaje de ontologías interpretable por un gestor semántico puede abrir nuevas posibilidades al nuevo paradigma de la gestión de redes.

5 CONCLUSIONES

- Esta Tesis se ha centrado en la representación de comportamiento de gestión de red, con el objetivo general de gestión red basada en ontologías y de proponer soluciones que permitan mejorar la interoperabilidad entre los modelos de información de gestión existentes, mediante la aplicación de ontologías como técnica de representación del conocimiento, para lo cual se analizaron los distintos modelos de gestión unificada existentes, encontrando que dichos modelos generan heterogeneidad en el momento de la gestión de red.

- Se desarrolla un modelo que se aplica sobre dos niveles importantes cómo son el de acceso al medio y el de aplicación, así posibilitando gestionar los equipos y dispositivos y sus comunicaciones con los servidores, para encontrar y detallar los inconvenientes de administración, y así facilitar la resolución de fallos y la aplicación de políticas sobre las redes de computadores.

- La utilización de Sistemas MultiAgente facilita la incorporación de los modelos ontológicos, como medio eficaz para el transporte de la información.

- Al definir la arquitectura de un gestor que maneje las ontologías, y aproveche el modelo de información común y las reglas de correspondencia y de comportamiento, se realiza una gestión desde un punto de vista común y neutro a los dominios de gestión en los que se encuentran los recursos gestionados. A la vez, las restricciones definidas sobre la información podrían ayudar a automatizar las tareas de vigilancia y control de dicho gestor.

- Una ontología especifica una conceptualización o una forma de ver el mundo, por lo que cada ontología incorpora un punto de vista. La ontología sobre sistemas de gestión de red contiene definiciones que proveen el vocabulario que permiten incorporar el dominio a través de reglas.

REFERENCIAS BIBLIOGRAFICAS

[1] International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), Overview of TMN Recommendations, Recomendación M.3000, febrero de 2000.

[2] International Telecommunications Union — Telecommunications Standardization Sector (ITU-T), Information techonology – Open Systems Interconnection — Common Management Information Protocol: Specification, Recomendación X.711, octubre de 1997

[3] International Telecommunications Union — Telecommunications Standardation Sector (ITU-T), Information techonology – Open Systems Interconnection — Common Management Information Service, Recomendación X.710, octubre de 1997

[4] Juan I. Asensio, Víctor A. Villagrá, Jorge E. López de Vergara, Julio J. Berrocal. *Experiences with SNMP-based integrated management of a CORBA-based electronic commerce application*. Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99), Boston, Massachusetts, EE.UU. A. Mayo de 1999.

[5] Waldbusser, Remote Network Monitoring Management Information Base, IETF Request For Comments 2819, mayo de 2000

[6] Distributed Management Task Force, Inc., Desktop Management Interface Specification, Version 2.0.1s, DMTF Standard DSP0005, enero de 2003.

[7] Distributed Management Task Force, Inc., Master MIF, Version 020507, mayo de 2002.

[8] Distributed Management Task Force, Inc., Specification for the Representation of CIM in XML, Versión 2.1, DMTF Standard DSP0201, mayo de 2002.

[9] G. Pavlov, From Protocol-based to Distributed Object-based Management Architectures, en Proceedings of the 4th Workshop of the Open View University Association (OVUA'97), Madrid, España, abril de 1997.

[10] Francisco Valera, Jorge E. López de Vergara, José I. Moreno, Víctor A. Villagrá, Julio Berrocal, Communication and Management Experiences in an E-Commerce MAS-Based Environment, en Communications of the Association for Computing Machinery (CACM). Volume 44, Number 4, abril de 2001.

[11] Object Management Group, CORBA 3.0.2 Specification, Overview, OMG document formal/02-06-38, junio de 2002. CORBA

[12] International Telecommunication Union -Telecommunication Standardization Sector (ITU-T), Information technology -Open Systems Interconnection -Systems management overview, Recomendación X.701, agosto de 1997.

[13] Object Management Group, Naming Service Specification, OMG document formal/02-09-02, septiembre de 2002. version 1.2,
[14] Object Management Group, Notification Service Specification, 1.0.1, OMG document formal/02-08-04, agosto de 2002. version
[15] Object Management Group, Telecom Log Service Specification, version 1.1, OMG document formal/02-11-12, noviembre de 2002.
[16] International Telecommunication Standardization Sector (ITU-T), Recomendación Q.816, enero de 2001. Union -Telecommunication CORBA-based TMN services,
[17] Object Management Group, CORBA 3.0.2 Specification, IDL Syntax & Semantics, OMG document formal/02-06-39, junio de 2002.
[18] International Telecommunication Union -Telecommunication Standardization Sector (ITU-T), TMN guidelines for defining CORBA managed objects, Recomendación X.780, enero de 2001.
[19] Object Management Group, UMLTM document formal/02-04-01, abril de 2002. Profile for CORBATM , OMG
[20] Proyecto REWERSE: Reasoning on The Web with Rules and Semantics, 6° Programa Marco (FP6) en el área IST (Information Society Technologies). [http://rewerse.net/]
[21] Distributed Management Task Force, Inc., Specification for CIM Operations over HTTP, Version 1.1, DMTF Standard DSP0200, enero de 2003.
[22] Distributed Management Task Force, Inc., and WBEM Solutions, Inc., DMTF Tutorial, [http://www.wbemsolutions.com/tutorials/DMTF/ dmtftutorial.pdf]
[23] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. T. Berners-Lee, Hypertext Transfer Protocol --HTTP/1.1, IETF Request For Comments 2616, junio de 1999
[24] H. Nielsen, P. Leach, S. Lawrence, An HTTP Extension Framework, IETF Request For Comments 2774, febrero de 2000.
[25] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, octubre de 2000.
[26] Distributed Management Task Force, Inc., Common Information Model (CIM) Core Model, Version 2.4, DMTF Whitepaper DSP0111, agosto de 2000.
[27] Distributed Management Task Force, Inc., Common Information Model Specification, Version 2.2, DMTF Standard DSP0004, junio de 1999.
[28] R. Studer, V.R. Benjamins, D. Fensel, Knowledge Engineering: Principles and Methods, en Data & Knowledge Engineering. 25: 161-197, 1998.
[29] Gregorio Fernández Fernández, Representación del conocimiento en sistemas inteligentes, Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, octubre de 2001. Ciberlibro disponible en http://www.gsi.dit.upm.es/~gfer/ssii/rcsi/.
[30] Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, en Scientific American, mayo de 2001.

[31] OntoWeb Consortium, Ontology Language version Standardisation Efforts version 1, Deliverable 4.0, enero de 2002.
[32] OntoWeb Consortium, Technical Roadmap, version 1.0˙ Deliverable 1.1, noviembre de 2001.
[33] J. E. López de Vergara, Director: V. Villagrá: Especificación de Modelos de Información de Gestión de Red Integrada Mediante el Uso de Ontologías y Técnicas de Representación del Conocimiento. Tesis Doctoral (Ph. D. Thesis), Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid. 2003.
[34] Guía Breve de Web Semántica http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica
[35] D. C. Fallside, P. Walmsley, XML Schema Part 0: Primer Second Edition, W3C ecommendation, 28 de octubre de 2004
[36] Peter D. Karp, Vinay K. Chaudhri, Jerome Thomere, XOL: An XML-Based Ontology Exchange Language, Technical Report, Artificial Intelligence Center, SRI International, agosto de 1999.
[37] F. Manola, E. Miller: RDF Primer, W3C Recommendation (10 February 2004).
[38] D. Beckett, RDF/XML Syntax Specification (Revised), W3C Recommendation, 10 de febrero de 2004
[39] D. Brickley, R. Y. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, 10 de febrero de 2004
[40] M. K. Smith, C. Welty, D. L. McGuinness: OWL Web Ontology Language Guide. W3C Recommendation, (February 2004)
[41] Z. Guessoum and J.P. Briot. From active objects to autonomous agents. IEEE Concurrency, 7(3):68-76, 1999.
[42] P. Maes. Agents that Reduce Work and Information Overload. In Jeffrey M. Bradshaw, editor, Software Agents. AAAI Press/MIT Press, 1997.
[43] Don Gilbert. Intelligent Agents: The Right Information at the Right Time, 1997.
[44] H. S. Nwana. Software Agents: An Overview. Knowledge Engineering Review, 11-2:205–244, 1995.
[45] MASIF-RTF Results. Technical report, Object Management Group, 1998.
[46] V. Honavar. Intelligent Agents and Multi-Agent Systems. IEEE Conference on Evolutionary Computation (CEC), 1999.
[47] FIPA ACL Message Structure Specification. Technical Report XC00061E, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, Agosto 2001.
[48] Yannis Labrou, Tim Finin, and Yun Peng. The current landscape of Agent Communication Languages . Intelligent Systems, 14(2), Marzo/Abril 1999.
[49] Venu Vasudevan. Comparing Agent Communication Languages. Technical report, Object Services and Consulting, Inc, Julio 1998.
[50] Tim Finin, Jay Weber, Gio Wiederhold, Mike Genesereth, Don McKay, Rich Fritzson, Stu Shapiro, Richard Pelavin, and Jim McGuire. Specification of the KQML Agent-Communication Language, 1993.
[51] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML – A Language and Protocol for Knowledge and Information Exchange. In International Conference on

Building and Sharing of Very Large-Scale Knowledge Bases, Tokyo, Diciembre 1993.
[52] FIPA Communicative Act Library Specification. Technical report, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS,SC00037J, Diciembre 2002.
[53] FIPA Agent Management Specification. Technical Report XC00023H, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, Octubre 2001.
[54] FIPA Request Interaction Protocol Specification. Technical Report SC00026H, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, Diciembre 2002.
[55] FIPA Agent Message Transport Service Specification. Technical Report SC00067F, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, Diciembre 2002.
[56] FIPA Iterated Contract Net Interaction Protocol Specification. Technical Report XC00030F, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, Agosto 2001.
[57] FIPA Proposal for Borda Count Interaction Protocol. Technical Report fin-00092, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, Diciembre 2002.
[58] J. E. López de Vergara, Director: V. Villagrá: Especificación de Modelos de Información de Gestión de Red Integrada Mediante el Uso de Ontologías y Técnicas de Representación del Conocimiento. Tesis Doctoral (Ph. D. Thesis), Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid. 2003.
[59] Casteleiro, Antonio Guerrero, Especificación del comportamiento de gestión de red mediante ontologías, Tesis Doctoral (Ph. D. Thesis), Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid. 2007
[60] International Telecommunication Union Standardization Sector (ITU-T), TMN Recomendación M.3400, febrero de 2000. -Telecommunication management functions,
[61] Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair, Integrated Management of Networked Systems. Morgan Kaufmann, 1999.
[62] WordReference.com Online Language Dictionaries. http://www.wordreference.com/es/en/frames.aspx?es=computadora
[63] ¿Que son los Dispositivos de Red? https://darkub.wordpress.com/2008/01/16/%C2%BFque-son-los-dispositivos-de-red/
[64] Qué es un servidor? www.anerdata.com/que-es-un-servidor.html
[65] Que es y para qué sirve un firewall? http://geekland.eu/que-es-y-para-que-sirve-un-firewall/
[66] Qué es un 'router' http://es.ccm.net/faq/2757-que-es-un-router
[67] WordReference.com Online Language Dictionaries. http://www.wordreference.com/es/en/frames.aspx?es=puertos
[68] Object Management Group, Unified Modeling Language (UML), version 1.4, Object Constraint Language Specification, OMG document formal/01-09-77, septiembre de 2001.

[69] Object Management Group, CORBA 3.0.2 Specification, General Inter-ORB Protocol, OMG document formal/02-06-51, junio de 2002.
[70] Uso de Protége http://www.dsi.fceia.unr.edu.ar/downloads/iia/recursos/Protege.pdf