

Definición del comportamiento de gestión de red con reglas SWRL en un marco de gestión basado en ontologías en OWL

Antonio Guerrero¹, Víctor A. Villagrà¹, Jorge E. López de Vergara².

¹Dpto. de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid (DIT-UPM).

²Dpto. de Ingeniería Informática, Universidad Autónoma de Madrid.

antonio.guerrero@telefonica.es, villagra@dit.upm.es, jorge.lopez_vergara@uam.es

Abstract. *The goal of the ontology-based management is the improvement of current network management interoperability techniques through the study and application of formal ontologies. Prior research work has analyzed different aspects of the problem: a study of the semantic expressiveness of current management information definition languages, the application of ontology languages (DAML+OIL and OWL) to represent the management information definitions, and the mapping and merging processes to obtain a semantic integration of those definitions. The usage of an ontology language allows additional advantages: these ontologies can include the definition of behavior rules in order to reason with them. Thus, usual behavior definitions included implicitly in the management information definitions and explicitly in policy definitions can then be expressed formally, and included with those definitions, which allows a manager reasoning and working with them. This paper focuses in the definition of behavior rules in management information with SWRL, a rule language defined to complement OWL functionality.*

1 Introducción

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 permiten acometer la gestión de un entorno heterogéneo utilizando un mismo mecanismo de comunicaciones y acceso a la información de gestión. Inicialmente, surgieron dos modelos (modelo de gestión Internet y modelo de gestión OSI) y, aunque el modelo de gestión Internet ocupó un gran ámbito de actuación, el modelo de gestión OSI todavía es utilizado en algunos escenarios sobre todo orientados a la gestión de redes de telecomunicación con la arquitectura TMN.

Adicionalmente, a finales de la década de los 90, surge un nuevo modelo que utiliza los protocolos de la Web para acometer los mismos objetivos de gestión integrada (modelo WBEM – *Web Based Enterprise Management*, Gestión de Empresa Basado en Web), si bien este modelo no excluía a los otros dos sino que permitía una cierta integración de distintos modelos de gestión de red integrada.

En este escenario de utilización de múltiples modelos de gestión de red integrada surge la misma problemática que se intentó solucionar con dichos modelos (la gestión propietaria), ya que un gestor tiene que interactuar con distintos recursos utilizando distintos mecanismos. Pero incluso así, el gestor no puede realizar una gestión verdaderamente integrada de su entorno, ya que no puede discernir acerca de la semántica de recursos definidos en distintos modelos para poder aplicar una política de gestión común a sus recursos, de forma independiente al modelo bajo el cual estén definidos.

Para solventar esta cuestión, en [1] se realiza una propuesta de la denominada Gestión Semántica basada 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 realizados en modelos diferentes. En dicha propuesta, se menciona otra ventaja que tiene esta aproximación, que es permitir integrar dentro de dicho modelo unificado la definición de aspectos de comportamiento del gestor y de los recursos, los cuales suelen estar mencionados como comentarios en las definiciones o explícitamente declarados pero fuera de las definiciones de los recursos gestionados.

Este artículo propone un punto de partido para dicha integración: comenzando con un modelo de información unificado y definido en un lenguaje de ontologías como es OWL, se propone la inclusión en dicho modelo de definiciones de comportamiento utilizando el Lenguaje de Reglas de la Web Semántica (SWRL, *Semantic Web Rule Language*). Para ello, el siguiente apartado presenta la arquitectura de gestión semántica sobre la que se va a trabajar, presentando a continuación el lenguaje SWRL. Dicho lenguaje permitirá la definición de comportamiento dentro de la información de gestión definida, como se comenta más adelante. Al mismo tiempo, se explican los tipos de comportamiento analizados: restricciones implícitas, comportamiento explícito del gestor y comportamiento explícito de los elementos gestionados. Para cada uno de estos tipos se aportan ejemplos de la aplicación de SWRL a la definición de comportamiento. Finalmente se resumen las ideas más relevantes.

2 Gestión Semántica

La arquitectura global propuesta en [2] 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 (SNMP, CIM, etc.) desde un punto de vista común y neutral a todos ellos. La Fig. 1 muestra la arquitectura propuesta para el gestor semántico.

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 (MIBs de SNMP, esquemas CIM, etc.) 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.

Las ontologías de correspondencia resultantes son utilizadas por los llamados “proveedores” o pasarelas de la Fig. 1 para traducir la información del gestor en el modelo unificado, a la información de los elementos de red en sus particulares modelos de gestión.

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 [3], 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 [4]. SWRL [5] extiende el conjunto de axiomas de OWL para incluir reglas condicionales (cláusulas de Horn), de tipo *si...entonces...*

Los axiomas y las reglas pueden ser utilizados en este marco de gestión para:

- 1) Restringir o definir de forma aún más precisa el comportamiento de la información de gestión en OWL. Esto permite asegurar un correcto uso e implementación de la información de gestión.

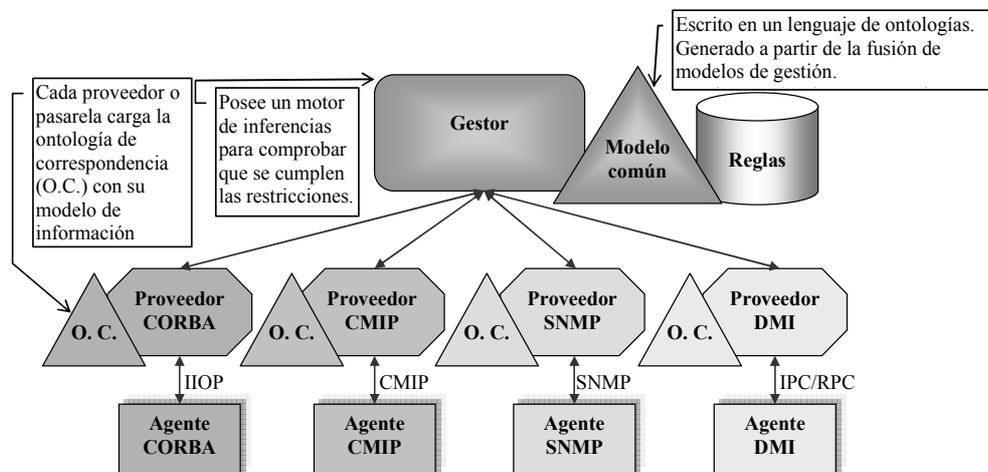


Fig. 1: Arquitectura del gestor semántico

- 2) Definir formalmente el comportamiento del gestor. Esto permite indicar lo que debería hacer el gestor si se cumplen determinadas condiciones en los elementos gestionados.
- 3) Definir formalmente el comportamiento de los elementos gestionados. Esto permite indicar lo que deberían hacer los recursos gestionados ante eventos determinados.

El objetivo de este trabajo es integrar la definición de la información de gestión (expresada en OWL) con la definición del comportamiento de gestión (expresado en SWRL). Para ello, el siguiente apartado presenta SWRL como un lenguaje que permite describir reglas para ontologías descritas en OWL.

3 SWRL: Definición de reglas para ontologías den OWL

Una ontología en OWL contiene una secuencia de hechos y axiomas. Los axiomas pueden ser de varias clases, p.e., axiomas de subclase, axiomas de equivalencia de clases, restricciones sobre propiedades. SWRL propone extenderlos con axiomas de regla.

Un axioma de regla consiste en un *antecedente* (cuerpo) y un *consecuente* (cabeza), cada uno de los cuales está compuesto por un conjunto (puede ser vacío) de *átomos*.

En la sintaxis inteligible por seres humanos de SWRL, una regla tiene la siguiente forma:

```
antecedente => consecuente
```

De manera informal, una regla puede interpretarse como una indicación de que si el antecedente es cierto, entonces el consecuente también es cierto. Utilizando esta sintaxis (notación lógica clásica), la propiedad de “ser tío de” se escribiría de la siguiente forma:

```
Persona(?x) & esPadre(?x,?y) &
esHermano(?y,?z) => esTio(?x,?z)
```

Una regla SRWL tiene por tanto la forma de una relación de implicación entre la *cabeza* y el *cuerpo*. La especificación de SWRL [5] ofrece una sintaxis abstracta que extiende la sintaxis abstracta de OWL descrita en [6] para incluir esta nueva relación en el lenguaje de ontologías. Las marcas XML que permiten describir estas reglas incluyen:

- `<ruleml:imp>`: Es el elemento que permite relacionar el *cuerpo* de la regla con la *cabeza*.
- `<ruleml:_body>`: Es el elemento que lista los *átomos* del *cuerpo* de la regla.

- `<ruleml:_head>`: Es el elemento que lista los *átomos* de la *cabeza* de la regla.
- `<ruleml:var>`: Permite definir las variables sobre las que evaluar las reglas.
- `<swrlx:individualPropertyAtom>`: Permite definir átomos referidos a propiedades concretas. También es posible definir átomos referidos a clases, rangos de datos, propiedades valuadas, o funciones propias de tipo matemático, cadenas de caracteres o fechas.

Como se ve, muchas de estas marcas no están definidas dentro del espacio de nombres de SWRL, sino de RuleML [7], un lenguaje de reglas definido con anterioridad que se ha tomado como base en la definición de SWRL. SWRL aporta sobre todo la definición de los átomos y su integración dentro de una ontología escrita en OWL.

4 Especificación del Comportamiento de Gestión en OWL+SWRL

4.1 Tipos de comportamiento de gestión

Para poder especificar comportamientos de gestión resulta conveniente clasificar qué tipos pueden existir. En concreto, se han identificado tres tipos de comportamiento de gestión:

- 1) Restricciones y reglas de comportamiento implícitas a los objetos modelados en las MIBs y esquemas CIM.
- 2) Comportamiento explícito del gestor, según la arquitectura tradicional gestor-agente en el que se indica como se comporta el gestor al obtener y analizar información de los agentes.
- 3) Políticas definidas explícitamente para especificar el comportamiento o configuración dinámica de los recursos gestionados (Gestión Basada en Políticas o *Policy-Based Management* [8])

En este marco de gestión semántica, las políticas se pueden definir en el mismo lenguaje de gestión, OWL+SWRL, que los objetos definidos en la base de información de gestión (MIB), con la ventaja que supone el trabajar con un modelo unificado.

Volviendo a la arquitectura de gestión propuesta en la Fig. 1, las definiciones de restricciones, reglas y políticas se almacenarían en la base de información de reglas (parte integrante del modelo común), cuyo propósito es verificar la integridad de la información, y automatizar el control de los elementos gestionados.

Los siguientes apartados se enfocan en cada uno de los tipos de comportamiento identificados.

4.2 Restricciones implícitas sobre la información de gestión

Este tipo de reglas se refieren a restricciones en el tipo de datos, cardinalidad o acceso. Son restricciones típicas sobre las propiedades y clases de los objetos gestionados. En este caso, las reglas en SWRL permiten complementar a los mecanismos ya existentes en OWL para realizar las definiciones de información de gestión, de forma que permiten expresar restricciones más complejas: valores que dependen del valor de otras variables, relaciones entre objetos, comportamiento de máquina de estados de los valores, etc.

En general, SWRL permite representar formalmente restricciones de comportamiento que se expresen en lenguaje natural de forma condicional (si ... entonces ...). Esto incluiría, entre otros, dependencia de valores, comportamiento de estados, comportamiento temporal o tipos complejos como funciones compuestas. A continuación se presentan, algunos ejemplos de este tipo de comportamiento encontrados en MIBs SNMP y cómo se aplica SWRL para definir este tipo de restricciones. En muchos casos la definición de las restricciones requerirá de la creación de nuevas clases y propiedades que extenderían la ontología de gestión.

Ejemplo 1

El primer ejemplo muestra cómo especificar con SWRL que el valor de una variable dependa del valor de otra. Para ello, se hace uso de una definición recogida de la MIB II de SNMP, relativa a la máscara de una dirección de destino en una tabla de encaminamiento:

```
ipRouteMask OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "... If the value of the
        ipRouteDest is 0.0.0.0 (a default
        route), then the mask value is also
        0.0.0.0. ..."
    ::= { ipRouteEntry 11 }
```

Esta restricción implícita se expresa en lenguaje natural en la cláusula DESCRIPTION, pero no se define formalmente en SMIV2, por lo que un gestor no puede implementarla de manera automática al compilar la MIB.

Suponiendo que la MIB II hubiese sido traducida e integrada en la base de información de gestión en OWL, la regla SWRL que definiría esta restricción sería la siguiente:

```
ipRouteEntry(IR?) ^ swrlb:equal
(ipRouteDest(IR?), "0.0.0.0")
=>swrlb:equal (ipRouteMask(IR?),
"0.0.0.0")
```

donde *ipRouteDest* e *ipRouteMask* son propiedades de la clase *ipRouteEntry*.

Cabe señalar que esta restricción aquí definida de forma simple, no es posible expresarla en otros lenguajes de definición de información de gestión. De hecho, el SMIng Working Group del IETF propuso el siguiente objetivo para la siguiente generación del lenguaje SMI, incluido en [9]: “*SMIng should provide mechanisms to formally specify constraints between values of multiple attributes*”, pero su desarrollo fue desestimado: “*This objective as is has been rejected as too general, and therefore virtually impossible to implement*”.

Por otro lado, y reforzando la idea en estudio, ocurre que tampoco es posible expresar esta restricción en OWL sin SWRL: una restricción sobre una propiedad (cláusula *owl:restriction*) no puede hacerse depender del valor de otra propiedad del mismo objeto.

Ejemplo 2

Este segundo ejemplo, obtenido también de la MIB II de SNMP, puede servir para mostrar la definición del comportamiento de estados en SWRL. En este caso se toma la columna de la tabla de conexiones de TCP que indica el estado de cada conexión.

```
tcpConnState OBJECT-TYPE
    SYNTAX INTEGER {
        closed(1),
        listen(2),
        synSent(3),
        synReceived(4),
        established(5),
        finWait1(6),
        finWait2(7),
        closeWait(8),
        lastAck(9),
        closing(10),
        timeWait(11),
        deleteTCB(12)
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "...
    ::= { tcpConnEntry 1 }
```

Como este parámetro es de tipo read-write, se trata entonces de definir formalmente en SWRL la máquina de estados TCP que se ilustra en la Fig. 2 para forzar al gestor a cumplirla.

En el modelo de información se ha definido una clase de objeto gestionado llamada *tcpConnEntry* con la propiedad relativa al estado actual de la conexión *tcpConnState*, y las propiedades auxiliares

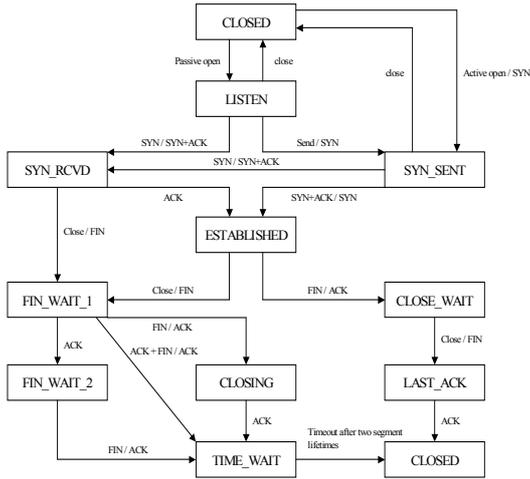


Fig.2: Máquina de estados TCP

tcpConnPreviousState, *tcpConnNextState*, que indican la lista de estados previos y siguientes, respectivamente. Un ejemplo de una de las reglas que permitirían definir la máquina de estados es el siguiente:

```

tcpConnEntry(cx?) ∧
swrlb:equal(tcpConnState(cx?),
"fin_wait_1")
⇒ swrlb:member(tcpConnNextState(cx?),
nextStatesForFin_Wait_1_List)

```

donde *nextStatesForFin_Wait_1_List* sería una lista (*rdf:list*) con los valores “closing”, “time_wait” y “fin_wait_2”.

Ejemplo 3

El tercer ejemplo muestra un caso de restricción temporal, tomando para ello la columna de la tabla de interfaces de red que indica el último cambio de una interfaz. Este ejemplo de restricción temporal podría ser desarrollado en SWRL si se hace uso de otras clases y propiedades. A continuación se muestra la definición de la propiedad anteriormente comentada, donde se ha marcado en **negrita** la cláusula que podría representarse con SWRL.

```

ifLastChange OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The value of sysUpTime at the
    time the interface entered its current
    operational state. If the current state
    was entered prior to the last re-
    initialization of the local network
    management subsystem, then this object
    contains a zero value."
 ::= { ifEntry 9 }

```

En este caso, tomando la propiedad *sysUpTime* de la clase *system*, se podría hacer dicha comparación temporal.

```

system(?x) ∧ ifEntry(?y) ⇒
swrlb:lessThan(ifLastChange(y?),
sysUpTime(?x))

```

Ejemplo 4

Un ejemplo de función compuesta a representar en SWRL puede ser un indicador sobre un determinado umbral, por ejemplo de temperatura (*flagTemp*):

```

Temperatura(t?) ∧ swrlb:lessThan(t?, 40) ⇒
flagTemp(ft?) ∧ swrlb:equal(valor(ft?), 0)
Temperatura(t?) ∧
swrlb:greaterThanOrEqual(t?, 40) ⇒
flagTemp(ft?) ∧ swrlb:equal(valor(ft?), t?)

```

Este indicador valdría 0° por debajo de 40°, y tomaría el valor de la temperatura en 40° o por encima.

4.3 Comportamiento explícito del gestor

El comportamiento del gestor ante el cumplimiento de determinadas condiciones en la red o los sistemas, puede especificarse por medio de reglas condicionales del tipo **if condición then acción**, que en SWRL se convierten en

condición ⇒ acción

o, ampliando la definición

conjunto de condiciones ⇒ conjunto de acciones

Acciones

Una de las carencias que posee OWL es la definición de funciones o métodos, habituales en lenguajes de información de gestión tales como CIM o GDMO. Para solventar este problema, las acciones en OWL pueden ser representadas mediante la ontología de servicios OWL-S (*OWL-Services*) [10]. Dicha ontología define entre otras cuestiones la clase *Process* que puede ser utilizada para este fin. Si se hace el símil de que ejecutar una acción puede realizarse “llamando a un servicio” que ejecuta la acción, entonces se puede definir esa acción como un *proceso OWL-S*, y crear una instancia de la clase *perform* con dicho *proceso* como argumento:

Perform(MyProcess)

La clase *Perform* es una clase auxiliar definida para representar la ejecución de procesos atómicos dentro de un proceso compuesto.

Ejemplos: *Perform(Reset)*,
Perform(SendAlarm(...)), *Perform(SetIPRoute(...))*

Estos procesos pueden ser nuevas acciones que se definan, o bien ser el resultado la integración de las operaciones ya existentes en los modelos de información traducidos (procedimientos CIM, operaciones de tipo "set" de SNMP, etc.)

En la arquitectura gestor-agente propuesta en la Fig. 1, el gestor invocaría las llamadas a los servicios que ofrecerían los proveedores, y estos a su vez ejecutarían las operaciones correspondientes sobre los elementos gestionados. El siguiente ejemplo muestra este tipo de definición de comportamiento del gestor semántico ante determinadas condiciones en la red.

Ejemplo 5

Este ejemplo toma la clase *CIM_SystemDevice* del esquema CIM, que posee dos ejemplares asociados a dos puertos, y activa uno si el otro no funciona, siguiendo la regla:

If LogicalPort #1 is "Operatively Down", then enable LogicalPort #2

En SWRL:

```
CIM_SystemDevice(LP1?) ^
swrlb:equal(deviceName(LP1?), "Lport1") ^
CIM_SystemDevice(LP2?) ^
swrlb:equal(deviceName(LP2?), "Lport2") ^
swrlb:equal(StatusInfo(LP1?),
"OPERATIVELY_DOWN")
⇒ Perform(SetAdminAvailability(LP2?,
"ENABLE"))
```

De esta forma se aplica una regla, no sobre toda una clase de objetos, sino sobre determinados objetos (ejemplares de una clase).

Al igual que ocurría con las restricciones implícitas, tampoco este tipo de definiciones explícitas podrían representarse en OWL sin SWRL como una restricción a los valores de una propiedad que dependan del valor de otra propiedad. A diferencia del ejemplo 1 del apartado anterior, en este caso se relacionan propiedades de objetos diferentes, que son ejemplares de una misma clase.

4.4 Comportamiento explícito de los elementos gestionados: aplicación a la gestión basada en políticas

Si las reglas que se definen en el modelo de información no son reglas para el comportamiento del gestor, sino reglas que definen el comportamiento de los elementos gestionados, entonces se necesita una arquitectura que permita distribuir las reglas o políticas a dichos elementos gestionados. Este tipo de arquitecturas son las que se definen en el entorno de

PBM (*Policy-Based Management*) o gestión de red basada en políticas. En este trabajo se hará referencia a los siguientes elementos de la arquitectura PBM propuesta por el IETF/DMTF [7]:

- Dispositivos PEP (Policy Enforcement Point): son aquellos elementos que pueden aplicar o ejecutar las políticas.
- Dispositivos PDP (Policy Decision Point): actúan de intermediarios entre los PEPs y el repositorio de políticas, siendo los responsables de interpretar las políticas del repositorio y comunicar las acciones a los PEPs.
- Repositorio de políticas: almacena las políticas definidas, que serán distribuidas a los PDPs

Una arquitectura basada en políticas presenta las siguientes características clave:

- Centralización: la definición del comportamiento se realiza en un único punto y puede ser distribuida de forma masiva a la red en lugar de definirla y aplicarla para cada elemento de forma separada
- Niveles de abstracción: se pueden definir políticas a diferentes niveles: alto nivel (p.e. reglas de negocio), niveles intermedios (p.e. definiciones en el nivel de servicio), y bajo nivel (p.e. políticas que aplican los elementos de red). Se prevé por tanto la necesidad de traducciones entre niveles para convertir las reglas de alto nivel en las políticas de bajo nivel que aplicarán los elementos de red, para lo cual resulta útil disponer de modelos (modelos de negocio, modelos de servicio, modelos de red). Los lenguajes de definición de políticas existentes pueden estar más orientados hacia la definición de políticas en unos u otros niveles. Como ejemplos citaremos PONDER y RBAC que son lenguajes que llamaríamos de "alto nivel" porque el modelo de información es muy cercano a la forma de pensar del ser humano, mientras que PCIM y PIB de COPS-PR permiten expresar políticas de muy bajo nivel, más cercanas a los lenguajes de gestión de los elementos de red.

A continuación se realiza una primera aproximación a la posibilidad de utilizar ontologías en OWL+SWRL como lenguaje para la definición de estos tipos de políticas, mediante los ejemplos siguientes.

Ejemplo 6: política de alto nivel en PONDER

En este ejemplo se define en SWRL una política ya definida en el lenguaje de definición de políticas PONDER, extraída de [11].

```
type rel ReportingT (ProjectManagerT pm,
SecretaryT secr) {
```

```

inst oblig reportWeekly {
    on timer.day ("monday") ;
    subject secr ;
    target pm ;
    do mailReport() ;
}
// . . . other policies
}

```

Esta política obligatoria llamada *reportWeekly* especifica que el sujeto con rol *SecretaryT* debe enviar por correo un informe cada lunes al sujeto con rol *ProjectManager*. En SWRL se podría expresar de la siguiente manera:

```

ProjectManagerT(pm?) ^ SecretaryT(secr?) ^
Timerday(t?) ^ swrlb:equal (t?, "monday")

⇒ Perform(mailReport(secr?, pm?))

```

De esta forma, otros tipos de políticas de PONDER u otros lenguajes de definición de políticas como PCIM y COPS podrían ser definidas en SWRL: políticas de autorización, de obligación, de filtrado, etc.

Ejemplo 7: política de bajo nivel

A continuación se presenta una definición en SWRL de una política de bajo nivel que podría ser ejecutada directamente por un elemento de red.

La regla siguiente marcaría con "0100" el campo "ToS" (tipo de servicio) de cada paquete IP cuya dirección IP destino cumpliera la condición de pertenecer al rango indicado.

```

IPpacket(ippacket?) ^
InsideIPRange(DestAddress(ippacket?),
"192.168.1.0 - 192.168.1.255")

⇒ Perform(SetTOSlabel(ippacket?, "0100"))

```

Este ejemplo es tal vez excesivamente sencillo y de escasa utilidad, puesto que posiblemente realizando una correspondencia directa de la PIB (*Policy Information Base*), en caso de existir, o del lenguaje de definición de políticas de bajo nivel que maneje el propio elemento de red o PDP fuera la mejor forma de representar las políticas de bajo nivel en la ontología común de información de gestión. Para esto se abordaría la traducción e integración en la ontología común, por un lado de la lista de condiciones que disparan la política (<condition set>), y por otro de la lista de acciones (<action list>) que ejecuta la política. Para ello puede resultar útil la aplicación del mecanismo M&M (*Merge and Map*, Fusión y Correspondencia) [2] de traducción e integración de información de diferentes lenguajes de gestión a la ontología común. El resultado de la aplicación de este método permitiría que, a la inversa, pudiesen traducirse las políticas representadas en la

ontología en OWL+SWRL al lenguaje nativo que entiendan los PDPs.

Nótese que las políticas mostradas son políticas sin eventos del tipo "<condition set> then do <action list>". No existen eventos explícitos para la evaluación de las condiciones, sino que los agentes deben realizar esta tarea cuando existe algún evento implícito como, por ejemplo, el inicio de una sesión. Este sería el caso por ejemplo de PCIM, pero no de COPS y PONDER que sí permiten políticas del tipo *event-condition-action*.

La gestión basada en políticas también trata sobre la traducción de políticas entre diferentes niveles de abstracción, de forma que las políticas definidas en los niveles más altos (p.e. nivel de negocio y nivel de servicio) son traducidas a políticas de más bajo nivel (p.e. nivel de red y de sistemas). El hecho de que las definiciones de políticas de los distintos niveles en una misma ontología de gestión estén integradas en un mismo lenguaje (OWL+SWRL) puede facilitar la traducción de políticas entre niveles, aunque este estudio queda fuera del alcance del presente trabajo.

5 Conclusiones

Como se ha mostrado en estudios previos [1][2][4] los lenguajes de ontologías tales como OWL incluyen las construcciones necesarias para definir las características típicas de la información de gestión de red que pueden encontrarse en otros lenguajes de definición de gestión como SMI-SNMP, MOF-CIM, etc., y que permite traducir e integrar las definiciones de los distintos lenguajes desde una punto de vista semántico.

En este marco de gestión semántica, este trabajo ha mostrado como SWRL:

- 1) añade capacidad de expresión para definir restricciones y reglas para las definiciones de la información de gestión expresadas en la ontología común en OWL propuesta. SWRL permite expresar formalmente restricciones y reglas de definición del comportamiento más complejas que las que podían definirse en OWL, enriqueciendo y añadiendo mayor expresividad a las definiciones de gestión.
- 2) permite definir explícitamente el comportamiento del gestor, y de los objetos gestionados, en el mismo lenguaje de ontologías, OWL+SWRL, que se ha propuesto para las definiciones de la información de gestión. Esto incluye:
 - Acciones que ejecutará el gestor ante la detección de determinadas

condiciones en los elementos gestionados o en el propio gestor

- Acciones que ejecutarán los elementos de red o sistemas gestionados ante la existencia de determinadas condiciones (políticas que definen el comportamiento de los elementos gestionados)

No obstante quedaría aún pendiente la integración en este marco de gestión semántico, de otros mecanismos que presentan los lenguajes de definición de políticas tales como eventos de red, eventos temporales, identificación y clasificación de políticas, etc.

Por otro lado, existe el problema de definir formalmente el conjunto de restricciones que actualmente hay definidas de manera implícita o explícita. En este sentido puede ayudar el uso de herramientas que integren la recogida, fusión y correspondencia de información de gestión con la búsqueda mediante heurísticos de restricciones descritas en los campos de descripción (con alta probabilidad cadenas como “if*then”, “have to” o “must” formarán parte de una de estas restricciones), y su posterior definición en SWRL a partir de la información ya definida en OWL.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia a través del proyecto GESEMAN (TIC2002-00934).

Referencias

- [1] Jorge E. López de Vergara, Víctor A. Villagrà, Juan I. Asensio, Julio Berrocal, “Ontologies: Giving Semantics to Network Management Models”, IEEE Network, Vol. 17, No. 3, May/June 2003. ISSN 0890-8044.
- [2] Jorge E. López de Vergara, Víctor A. Villagrà, Julio Berrocal, “Benefits of Using Ontologies in the Management of High Speed Networks”, Lecture Notes in Computer Science, Vol. 3079, Springer-Verlag. ISSN 0302-9743
- [3] Michael K. Smith, Chris Welty, Deborah L. McGuinness, “OWL Web Ontology Language Guide”, W3C Recommendation 10 Feb. 2004.
- [4] Jorge E. López de Vergara, Víctor A. Villagrà, Julio Berrocal, “Applying the Web Ontology Language to management information definitions”, IEEE Communications Magazine, Vol. 42, Issue 7, July 2004, pp. 68-74. ISSN 0163-6804
- [5] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf, Mike Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”, W3C Member Submission 21 May 2004.
- [6] Peter F. Patel-Schneider, Patrick Hayes, Ian Horrocks, “OWL Web Ontology Language Semantics and Abstract Syntax”, W3C Recommendation 10 February 2004.
- [7] Rule Markup Initiative, <http://www.ruleml.org/>
- [8] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, S. Waldbusser, “Terminology for Policy-Based Management”, IETF Request For Comments 3198
- [9] C. Elliott, D. Harrington, J. Jason, J. Schoenwaelder, F. Strauss, W. Weiss, “SMIng Objectives”, IETF Request For Comments 3216, December 2001.
- [10] David Martin, editor, “OWL-S: Semantic Markup for Web Services”, W3C Member Submission 22 November 2004
- [11] Nicodemos Damianou, Naranker Dulay, Emil Lupu, Morris Sloman, “The PONDER Policy Specification Language”, Proc. Policy 2001: Workshop on Policies for Distributed Systems and Networks, Bristol, UK, 29-31 Jan. 2001, Springer-Verlag LNCS 1995, pp. 18-39.