

Modelo ontológico de verificación de sistemas multiagentes diseñados bajo Masina

Verification ontological model of multi-agent systems designed using Masina

José Aguilar¹, PhD, Reina Castellanos², Ing.

1. CEMISID, Departamento de Computación, Universidad de los Andes, 2. Distrito Tecnológico Petróleos de Venezuela. Mérida, Venezuela
aguilar@ula.ve, reinacastell@yahoo.com

Recibido para revisión: 22 de Septiembre de 2008, Aceptado: 28 de Noviembre de 2008, Versión final: 22 de Diciembre de 2008

Resumen—Existe un número importante de metodologías para la especificación y desarrollo de Sistemas Multiagentes (SMA), pero normalmente no garantizan que la especificación del sistema realmente satisfaga los requerimientos de diseño. Una de ellas es la metodología MASINA (MultiAgent Systems in Automation). Esta investigación apunta a presentar un Modelo Ontológico para la Verificación de los diseños de SMA hechos con MASINA, donde las propiedades a verificar son formalizadas semánticamente y son expresadas como sentencias en Lógica de Primer Orden. Para lograr esto se usan dos técnicas: la verificación de manera composicional (el SMA es visto desde diferentes niveles de abstracción, los cuales son llamados Componentes o Sub-componentes, y a los cuales se les deben verificar las relaciones entre ellos) y el cruce de modelos (MASINA posee un conjunto de modelos que son cruzados entre ellos para verificar si son consistentes sus relaciones). Dicho modelo ontológico ha sido implementado en Protégé-OWL y probado en un caso de Estudio (Diseño de un Sistema Operativo Web usando SMA).

Palabras Clave—Sistemas multi-agentes, MASINA, Verificación composicional, Verificación basada en cruce de modelos, Ontología, Verificación de diseños.

Abstract—An important number of methodologies for the specification and development of Multiagent Systems exist (MAS), but normally do not guarantee that the specification of the system really satisfies the design requirements. One of them is the MASINA methodology (MultiAgent Systems in Automation). This paper presents an Ontological Model for the Verification of the MAS designs using MASINA, where the properties to verify are formalized semantically and are expressed like First Order Logic sentences. In order to obtain this, two techniques are used: the Compositional Verification (the MAS is seen from different levels of abstraction, which are called Component or Sub-components, and to which the relations among them must be verified) and the crossing of models (MASINA has a set of models that must be

crossed in order to verify if their relations are consistent). This ontological model has been implemented in Protégé-OWL and has been tested in a case of Study (Design of a Web Operating system using MAS).

Keywords—Multi-agent systems, MASINA, Compositional verification, Verification based on crossing of models, Ontology, Designs verification

I. INTRODUCCIÓN

Al proceso de comprobación que asegura que el software del usuario se lo conoce con el nombre genérico de *Verificación*. La comprobación puede ser un proceso muy complejo, tanto en modo conceptual como en el computacional [5]. En general, existe un buen número de metodologías para desarrollar SMA, pero muy pocas de ellas tienen componentes que permitan la verificación de sus diseños.

Por otro lado, la Metodología MASINA (*MultiAgent Systems in Automation*) surge como una extensión de MAS-CommonKADS, a los fines de adecuar la metodología al modelado de procesos de automatización industrial e integrarle la noción inteligente, entre otras cosas [1, 16]. MAS-CommonKADS es una metodología que permite la especificación de SMA basados en mecanismos de comunicación predeterminados y en la teoría de pase de mensajes. Los agentes diseñados con MASINA deben ser verificados de acuerdo a ciertos criterios de evaluación, para garantizar que el SMA funcione apropiadamente. Básicamente, en este trabajo nosotros proponemos un modelo ontológico de verificación basado en dos técnicas: la verificación de manera composicional y el cruce de modelos.

Algunos trabajos de verificación composicional son los siguientes: En [10] es introducido un método de verificación composicional con especificaciones formales para SMA. En ese trabajo las propiedades de un nivel de abstracción del SMA se establecen por las suposiciones que se realizan de las propiedades de los agentes que lo componen, y así sucesivamente. En [5] se usa la lógica epistémico temporal para formalizar el proceso de verificación composicional para SMA. En ese trabajo las propiedades para verificar son formalizadas semánticamente en términos de modelos epistémicos temporal (lenguaje que permite expresar cambios en el tiempo). En [13] se presenta una comprobación composicional que se usa para el análisis formal de un SMA para negociación uno-a-muchos. Otros trabajos son [11, 12].

Por otro lado, un trabajo muy vinculado al nuestro en el área de métodos de verificación basado en cruce de modelos ha sido propuesto en [9]. En dicho trabajo se propone un método de verificación que es una mezcla de la Verificación Formal y el Cruce de Modelos. Para la verificación a nivel Macro, cada vez que un nivel de abstracción es descompuesto, los nuevos componentes deben ser validados de acuerdo a sus relaciones. A nivel micro, la verificación es hecha a través del cruce de modelos, es decir, se compara cada modelo con el resto de los modelos.

Inspirado en [9], en este trabajo desarrollamos un modelo ontológico para la verificación de SMA, el cual mezcla las dos técnicas anteriormente indicadas. Así, planteamos un *Método para la Verificación de los diseños de los SMA hechos con MASINA*, el cual usa sentencias escritas en Lógica de Primer Orden para describir la formalización semántica de las propiedades a verificar. Además, se implementa dicho modelo en Protégé-OWL [14], y se prueba en un caso de estudio de un Sistema Operativo Web basado en SMA [6].

II. ASPECTOS TEÓRICOS

A. MASINA

La metodología MASINA (*MultiAgent Systems in Automation*), es una extensión a MAS-CommonKADS [1, 2, 16], a los fines de adecuar la metodología al modelado de procesos de automatización industrial, aunque puede ser usada en otros contextos. MASINA ofrece un conjunto de modelos que agrupan los elementos básicos y necesarios para describir al SMA. Los modelos están basados en la identificación de atributos y sus relaciones; el producto generado consiste en una serie de plantillas que describen al SMA. Básicamente, está constituida por cinco modelos, los cuales describen un SMA:

a. Modelo de agente: Este modelo describe cada uno de los agentes del SMA, sus *objetivos* (responsabilidad asignada o adoptada por un agente), *características*, *habilidades*, *capacidades* (requisitos para poder cumplir los objetivos asignados), *restricciones* (normas, preferencias y permisos

que debe seguir el agente), y los *servicios* que ofrece a otros agentes. Además, este agente tiene otros atributos como: *componentes de ese agente* (si él mismo es un SMA) y *marco de referencia* que sirve para describir a ese agente, es decir, si ese agente está basado en algún modelo de referencia de SMA, tal como el SCDA [15].

b. Modelo de tareas: En este modelo se especifican las tareas que deben desarrollar los agentes, es decir, se describen las tareas que se deben realizar en el entorno organizacional. Cada tarea consta de *capacidades* (visión de más alto nivel de la competencia necesaria para ejecutar la tarea), *ingredientes de entrada y de salida* (Representación de la entrada y salida de una tarea), y un *entorno de operación* (restricciones formales sobre la ejecución de la tarea, establecidas en la organización). Además, este modelo tiene atributos para especificar las *propiedades de las tareas que requieren el uso de técnicas inteligentes*, y para describir el *procedimiento que se debe seguir para la ejecución de la tarea*.

c. Modelo de Inteligencia: El modelo de inteligencia está compuesto por un conjunto de elementos que están en correspondencia con el razonamiento humano. Especifica los conocimientos y métodos de razonamiento necesarios para que los agentes inteligentes cumplan sus objetivos, así como sus mecanismos de aprendizaje. Se propone un esquema que integra conceptos como *experiencia* (conocimiento adquirido por el agente durante su funcionamiento), *conocimiento de dominio* (involucra las variables que usa cada agente), *conocimiento estratégico* (información asociada al problema y derivada de las actividades hechas por otros agentes), *mecanismo de aprendizaje* (describe el proceso de acumulación de experiencia y el proceso para adaptar los mecanismos de razonamiento y actualizar el conocimiento estratégico), *conocimiento de tareas* (proporciona al agente información sobre las tareas y quien puede realizarlas), y *mecanismo de razonamiento* (utiliza el conocimiento adquirido para razonar e inferir/deducir soluciones. Por ejemplo, el uso de máquinas de aprendizaje o motor de inferencia del agente inteligente, entre otras). El componente inteligente modelado se percibe como un módulo que puede ser insertado y extraído de la estructura de cada agente, agregándole de esta forma capacidades inteligentes al agente, sin que esto comprometa la arquitectura de los agentes.

d. Modelo de Coordinación: El modelo de coordinación permite al usuario establecer el conjunto de estrategias que las comunidades de agentes usarán para el logro de los objetivos grupales. Estas estrategias, por lo general, imitan actitudes de coordinación de grupos humanos, tales como los procesos de contratación y de subastas, entre otros, donde la interacción y el intercambio de información generan acciones individuales que sumadas permiten alcanzar metas específicas. En otras palabras, este modelo describe los mecanismos que permiten realizar una conversación entre un grupo de agentes. Las interacciones presentes en los

agentes involucrados en una conversación dada se describen con los siguientes atributos: *Nombre* (nombre de la conversación), *Tipo de Conversación* (predefinida, emergente, entre otras), *Objetivo* (objetivos que satisface la conversación), *Agentes* (agentes participantes en la conversación), *Iniciador* (agente que inicia la conversación), *Servicio* (servicios proporcionados), *Actos de Habla* (actos de habla involucrados en la conversación), *Descripción* (descripción de la conversación en lenguaje natural), *Precondición* (condiciones que deben darse en el emisor y receptor para que la conversación pueda darse con éxito) y *Condición de Terminación* (condiciones en el emisor y receptor tras el desarrollo de la conversación). Además, en este modelo se especifica el *esquema de coordinación* (indica el tipo de coordinación usada: adaptativa o predefinida, objetivo a seguir, procedimiento usado en la coordinación adaptativa, y la planificación por defecto usada en la coordinación predefinida), *la planificación usada* (tipo y técnica de planificación usada), *el mecanismo de comunicación directa* (protocolos usados en la comunicación), *o de comunicación indirecta* (estrategia de actualización de estímulos y del umbral de respuesta en las comunicaciones), *el metalenguaje* (define el metalenguaje usado para la comunicación directa entre los agentes), y *la ontología* (vocabulario manejado).

e. Modelo de Comunicación: MASINA plantea que el modelo de comunicación describa los actos de habla involucrados en las conversaciones desarrolladas en el modelo de coordinación. Así, él describe todas las comunicaciones directas a través del intercambio de información usando mecanismos del tipo pase de mensajes, y comunicaciones indirectas a través del depósito de información en objetos (uso de estrategias de tipo memoria compartida y métodos de estímulo-respuesta), que reflejan las interacciones entre los agentes. Para representar eso, el modelo de comunicación se basa en la descripción de los actos de habla presentes en una conversación. Los atributos que permiten describir cada

acto de habla involucrado en una conversación son los siguientes: *objetivo* (objetivo del acto de habla), *Tipo* (asertivo, directivo, compromiso, expresivo o declarativo), *Agentes Participantes* (agentes que participan en el acto de habla), *Comunicación* (un acto de habla puede ser realizado de manera directa o indirecta.), *Emisor* (emisor del mensaje), *Receptor* (receptor del mensaje), *Conversación* (nombre de la conversación donde participa), *Servicio* (nombre del servicio asociado), *Datos Intercambiados* (descripción de los datos intercambiados en el acto de habla), *Descripción* (descripción del acto de habla en lenguaje natural), *Precondición* (condiciones que deben darse en el emisor y receptor para que el acto de habla pueda darse con éxito.), *Condición de Terminación* (condiciones en el emisor y receptor tras el desarrollo del acto de habla), *Performativa* (verbo que describe la intención del acto de habla), *Medio de Comunicación* (vía de comunicación).

B. Verificación de Sistemas Multiagentes

El propósito de la Verificación, es probar que bajo un cierto conjunto de suposiciones, un SMA satisface un cierto conjunto de propiedades.

1) SMA composicional

Un SMA composicional puede verse y especificarse en diferentes niveles de abstracción. El nivel superior, denotado por L_0 , es el SMA completo (componente S) donde información interna y procesos no son especificados en este nivel de abstracción (información y procesos son ocultos). En el próximo nivel de abstracción, L_1 , la estructura interna del sistema es tomado en términos de sus componentes, pero los detalles de los componentes son ocultos. El siguiente nivel mas bajo de abstracción es L_2 , y es donde se especifican los sub-componentes (ver Figura 1). Algunos componentes no pueden descomponerse en sub-componentes, esta son llamados *componentes primitivos*.

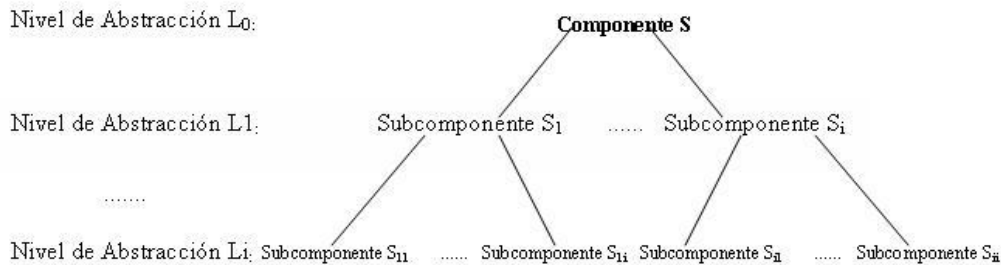


Figura 1. Niveles de abstracción de un SMA

El Método de verificación composicional basado en lógica [4] es una tendencia reciente en la literatura de verificación, el cual estudia el uso de la composición y abstracción para estructurar el proceso de Verificación. La verificación composicional apunta a manejar la complejidad del proceso

para explotar composicionalmente la arquitectura del sistema. Toma en cuenta la estructura composicional durante el proceso de verificación [5]. La verificación de un componente compuesto es hecho usando propiedades de los sub-componentes. Esto implica que propiedades de niveles

diferentes de abstracción son envueltas en el proceso de verificación. Estas propiedades tienen relaciones lógicas jerárquicas en el sentido en que cada nivel toma las especificaciones de los componentes como una propiedad lógicamente implícita (por conjunción) de las más bajas propiedades relacionadas en la jerarquía, como también de las relaciones lógicas entre las propiedades dentro de un mismo nivel de abstracción. El método de verificación composicional puede formularse informalmente como sigue:

A. Verificar un nivel de abstracción contra el otro

Para cada nivel de abstracción el procedimiento es el siguiente:

1. Determinar que propiedades son de interés para el componente D (nivel alto).
2. Determinar las propiedades de los componentes del nivel bajo que garantice las propiedades de la componente D.
3. Probar las propiedades del componente D en base a las propiedades de sus sub-componentes, usando la especificación del sistema que define como D esta compuesto de estas sub-componentes.

B. Proceso de verificación Total

Para verificar el sistema completo:

1. Determinar las propiedades que son deseables para el sistema entero.
2. Aplicar el procedimiento anterior iterativamente. En la iteración, las propiedades deseadas del nivel de abstracción L_i en estudio son:
 - Aquellas determinados en el paso 1, si $i=0$, ó
 - Las suposiciones hechas por el nivel superior, si $i>0$

2) Cruce de Modelos

Se valida y verifica el diseño de los SMA desde el punto de

vista de los modelos (plantillas generadas por la Metodología de desarrollo de Agentes). El Proceso de Verificación se hace a través de la comparación de cada modelo con el resto de los modelos, esto se conoce como **cruce de modelos**. El Cruce de Modelos permite comprobar si se cumplen las relaciones implícitas entre los diferentes Modelos generados por la Metodología de Agentes a usar, y la cantidad de cruces generados dependerá de la cantidad de plantillas (Modelos) establecidos por dicha Metodología.

III. VERIFICACIÓN PARAMASINA

El proceso de verificación del diseño obtenido con MASINA se realiza en dos niveles:

- **A nivel Macro (verificación composicional):** En éste nivel se verifica el diseño de cada subsistema considerando la estructura y las interacciones del sistema. Para ello, el sistema es visto desde diferentes capas de abstracción. En éste Nivel se debe comprobar que los componentes de la Capa L_{i+1} son los responsables de los objetivos de los componentes L_i , manteniendo las interacciones externas establecidas por los componentes de la Capa L_i . Así a través de un conjunto de criterios se comprobará si el diseño establecido en una capa superior, al ser descompuesto cumple o no con las especificaciones del diseño establecido en la capa inferior.
- **A Nivel Micro (cruce de modelos):** En éste nivel se verifica el diseño desde el punto de vista de los modelos de MASINA. La verificación se hace a través de la comparación de cada modelo con el resto de los modelos [1]. El Cruce de los Modelos de MASINA es mostrado en la Figura 2. En éste caso se determina si la relación que se establece entre los modelos es coherente. Esto se debe hacer para cada Agente diseñado y para cada componente definido en el SMA.

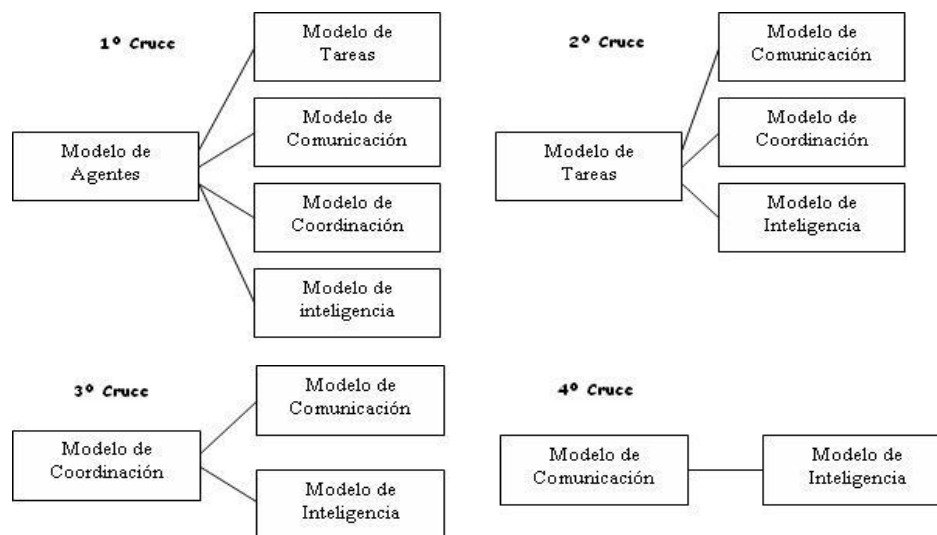


Figura 2. Cruce de los modelos en las plantillas diseñadas con MASINA

A. Criterios para la Verificación A Nivel Macro

Existen en total cinco criterios para realizar la verificación a nivel macro. Cada vez que una Capa de Abstracción L_i es descompuesta, los nuevos componentes de la Capa L_{i+1} deben ser validados de acuerdo a ciertos Criterios. El primer criterio especifica que: “Un Componente de la Capa L_{i+1} satisface uno o varios objetivos de la Capa L_i ” (Figura 3 muestra la sentencia lógica, las fuentes de información y el tipo de relación que se da en este criterio).

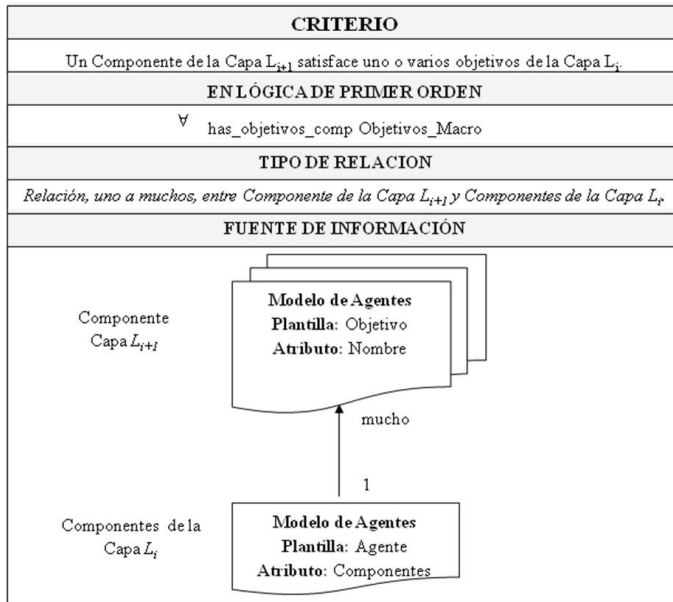


Figura 3. Criterio para la Verificación A Nivel Macro

El resto de los criterios son (ir a [8] para ver los detalles de estos criterios: sentencias lógicas, etc.):

- Un Componente de la Capa L_i esta compuesto por uno o varios componentes de la Capa L_{i+1} .
- Los Componentes del Nivel L_{i+1} son responsables de las tareas del Nivel L_i .
- Cada Agente del SMA es identificado con nombre único y no repetible en el SMA
- Toda Conversación de la Capa L_i debe reflejarse en el objetivo de algún componente L_{i+1}

B. Criterios para la Verificación A Nivel Micro

Como se dijo antes, la Verificación A Nivel Micro se realiza desde el punto de vista del Cruce de Modelos. Se compara cada Modelo con el resto de los Modelos, la misma se hace a través de un conjunto de criterios. Por ejemplo, para el primer cruce entre el Modelo de Agentes con el Modelo de Tareas, un criterio de comparación indica que: “Toda Tarea debe corresponder a un servicio de un Agente” (figura 4 muestra la sentencia lógica, las fuentes de información y el tipo de relación que se da en este criterio). El resto de criterios para este cruce son (ir a [8] para ver los detalles de cada uno):

- Para Hacer una Tarea se requiere el grado de capacidad de un Agente
- Los Actos de Habla son ejecutados por uno o varios agentes
- Los Actos de Habla son iniciados por uno o varios agentes
- Cada Agente debe conocer las Ontologías usadas en las comunicaciones donde participe

En cuanto al cruce entre el Modelo de Agentes con el Modelo de Coordinación, los criterios a considerar son (los detalles de cada uno son descritos en [8]):

- Una Conversación debe satisfacer el objetivo de uno o varios agentes
- Las conversaciones son iniciadas por uno o varios agentes

Y así sucesivamente, por cada cruce entre modelos, según los cruces establecidos en la figura 8, se van definiendo los respectivos criterios para realizar las verificaciones del caso. En total, incluyendo los criterios ya nombrados para los niveles macros y micros, son 21 criterios de verificación, cuyos detalles son presentados en [8].

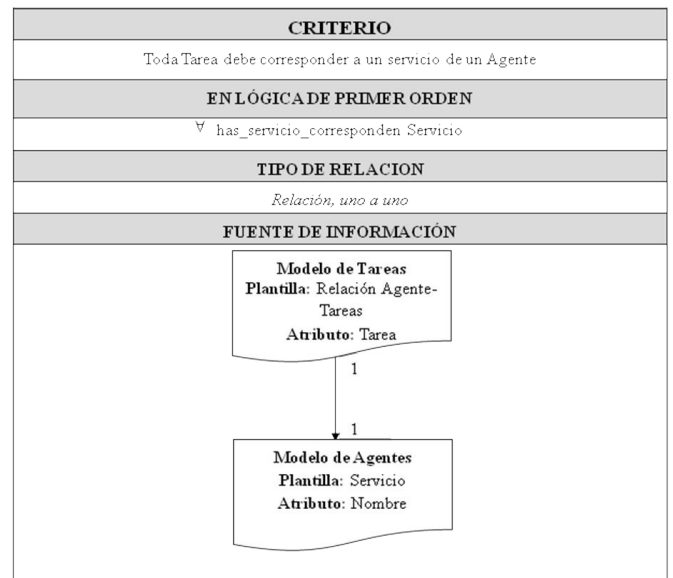


Figura 4. Criterio para la Verificación A Nivel Micro

IV. DESCRIPCIÓN DEL MODELO ONTOLÓGICO

La construcción del modelo Ontológico se basó en dos fases:

1. Adquisición del conocimiento: en ésta fase se especifican las clases (descripción de elementos que contienen individuos) que se usan a lo largo del desarrollo de la Ontología, así como también de las propiedades (relaciones que existen entre clases). Por ejemplo: Clase Persona: describe una clase que engloba las propiedades y características de una Persona. La Clase Persona posee varias propiedades o relaciones: tiene_nombre, tiene_apellido, tiene_Amigos, etc.

2. Especificación de restricciones (condiciones usadas en las relaciones, llamada también definiciones): en ésta fase se identifican y especifican detalladamente cada una de las restricciones que limitarán la manera en la que las relaciones pueden ser usadas. En nuestro caso, los criterios de verificación son descritos aquí. Por ejemplo: La Clase Persona es subclase de la Clase humano, además, tiene por lo menos un nombre, un apellido, puede tener uno a varios amigos, etc.

Esta Ontología ha sido desarrollada utilizando las herramientas Protégé-OWL versión 3.2, GraphViz versión 2.8. y Ontoviz versión 2.2 [14].

A. Adquisición del conocimiento

Como se puede observar en la Figura 5, las clases principales de la “Ontología del Sistema de Validación de Agentes diseñados con MASINA” es la clase SMA, de la cual se derivan la clase A_NIVEL_MACRO y la clase A_NIVEL_MICRO.



Figura 5. Grafo de la Ontología del Sistema de Validación de Agentes diseñados con MASINA

Con respecto a las propiedades, que representan la interacción y el enlace entre conceptos (clases), son descritas a continuación dos de ellas (sus nombres indican la relación que definen), una a Nivel Macro y otra a Nivel Micro, para el

resto ver [8] (la Figura 6 muestra el listado de todas). Las propiedades son descritas por dos variables, para indicar como se enlazan individuos de un *dominio* a individuos de un *rango*.

- **Propiedad A Nivel Macro:**

has_objetivos_comp: relación que indica la existencia de objetivos dentro de la Clase Componente.

Dominio: Clase Componente

Rango: Clase Objetivos_Macro.

- **Propiedad A Nivel Micro:**

has_servicio_corresponden: propiedad que describe la existencia de un Servicio.

Dominio: Clase MTAREAS-TAREAS

Rango: Clase Servicio

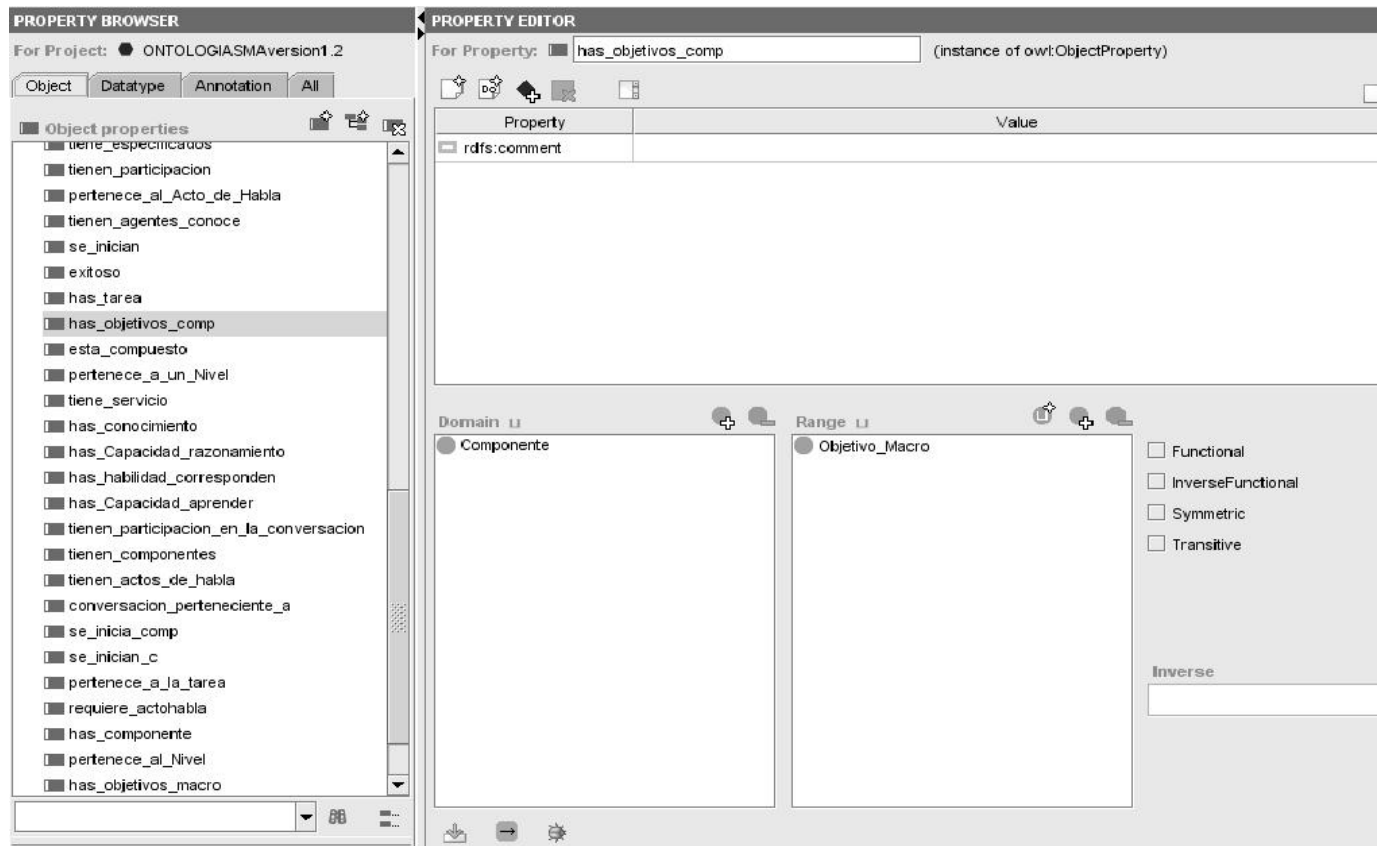


Figura 6. Propiedades en Protégé-OWL

B. Definición y Especificación de Restricciones

En ésta fase se realiza la identificación de las restricciones que limitarán la manera en la que las relaciones pueden ser formadas. En nuestro caso, aquí plasmamos las sentencias lógicas que describen los criterios de verificación previamente definidos.

- **A Nivel Macro:** En este nivel, por ser verificación composicional, debemos caracterizar los criterios de verificación a nivel de componentes. Los criterios en este caso son 5, y fueron definidos en III.A. En la figura 7 se describen los mismos, considerando que un **COMPONENTE** es una subclase de **A_NIVEL_MACRO**, esta compuesto a su vez por otros componentes, y tienen como propiedades adicionales Objetivos y Tareas. Además, pertenecen al menos a un Nivel de los Niveles de abstracción de un SMA. Veamos en detalle la descripción del primer criterio (ver Figura 3):

∇ has_objetivos_comp only Objetivos_Macro

Donde:

- has_objetivos_comp: relación definida en la sección IV.A.
- Objetivos_Macro: Clase Objetivos_Macro la cual esta formada por los objetivos a Nivel Macro.
- only indica que es un objetivo único de el.
- Necessary: indica que es un criterio necesario, pero no suficiente, para describir la relación entre componentes

El resto de criterios se describen de la misma forma en [8].

- **A Nivel Micro:** Recordemos que en este nivel la verificación se hace según el modelo de cruce de modelos, lo que significa que debemos describir los criterios de verificación específicos a este caso (ver sección III.B). Veamos el ejemplo del Cruce entre el Modelo de Agentes y el Modelo de Tareas, y sus 5 criterios de verificación para este caso (ver Figura



Figura 7. Definición de la Clase Componente en Protégé-OWL

8). **MTAREAS-TAREAS** es una Clase que indica este cruce, por lo que para ella se definen los 5 criterios de verificación, tales como *cumple_un_Objetivo* para indicar la existencia de Objetivos cumplidos, *es_realizada_por_Agente* que indica el agente que realiza la Tarea, y así el resto. Veamos en detalles la descripción del criterio de la Figura 4, el cual es el primer criterio de verificación de este cruce:

\forall **has_servicio_corresponden** Servicio

Donde:

- has_servicio_corresponden: relación definida en la sección 4.1.
- Servicio: Clase Servicio la cual esta formada por los servicios ofrecidos por los Agentes.
- only indica que es un objetivo único de el.
- Necessary: indica que es un criterio necesario, pero no suficiente, para describir la relación entre componentes



Figura 8. Definición de la Clase MTAREAS-TAREAS en Protégé-OWL

El resto de criterios se describen de la misma forma en [8].

V. CASO DE ESTUDIO: DISEÑO DE UN SISTEMA MANEJADOR DE COMUNIDADES Y REPOSITORIOS LOCALES

El caso de estudio que se analiza se basa en el trabajo [6], en el que se propone el Diseño de un Sistema Manejador de Comunidades para un Sistema Operativo Web. Este trabajo plantea la arquitectura de un Sistema Manejador de Repositorios Locales (SMRL) (encargado de gestionar los

repositorios locales) para un sistema Operativo Web (SOW). El SOW tiene como objetivo principal proveer una plataforma que permita a los usuarios beneficiarse del potencial computacional ofrecido en la Web, compartiendo recursos y resolviendo los problemas de heterogeneidad y adaptabilidad dinámicamente. El modelo fundamental del SOW esta conformado por cuatro subsistemas que llevan a cabo una serie de funciones coordinadas que permitirán un uso eficiente de los recursos sobre Internet, a pesar de sus características dinámicas, alta movilidad y homogeneidad (Ver Figura 9)

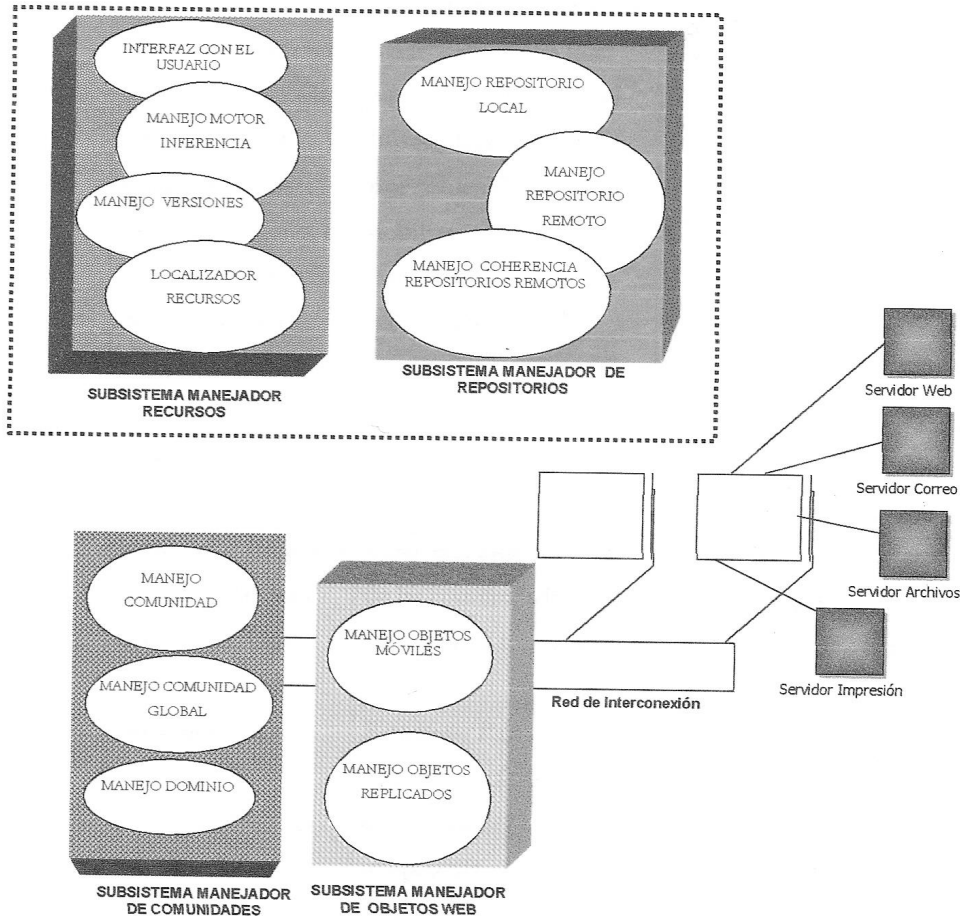


Figura 9. Subsistemas que integran el SOW

A. SOW Nivel Macro

El SOW es visto desde diferentes niveles de abstracción, debido a que es un SMA. Desde el nivel (capa) más alto, el SOW puede ser visto como un solo componente, denotado por L_0 , donde la información interna y los procesos en él no son especificados (se ocultan las comunicaciones y los procesos internos en el SMA). El próximo nivel de Abstracción, L_1 son los componentes S_i (Subsistemas) que cumplen con los objetivos del SOW (Subsistema Manejador de Comunidades-SMC, Subsistema Manejador de Repositorios, que a su vez se divide en dos: locales-SMRL y remotos SMRR, Subsistema Manejador

de Objetos Web-SMOW y Subsistema Manejador de Recursos-SMR). Finalmente, cada subcomponente S_i (o subsistema) puede verse como una colección de agentes que cumple con los objetivos de dicho subsistema; por ejemplo el SMRL se divide en cinco agentes (ver Figura 10), los cuales son: Agente Coordinador-COO, Agente Administrador de los Recursos Locales-ARL, Agente Administrador de la Información Local-AIL, Agente Administrador de las Trazas o uso de los objetos web-AT, y Optimizador-Opt. El resto de los subsistemas se subdividen también en diferentes agentes.

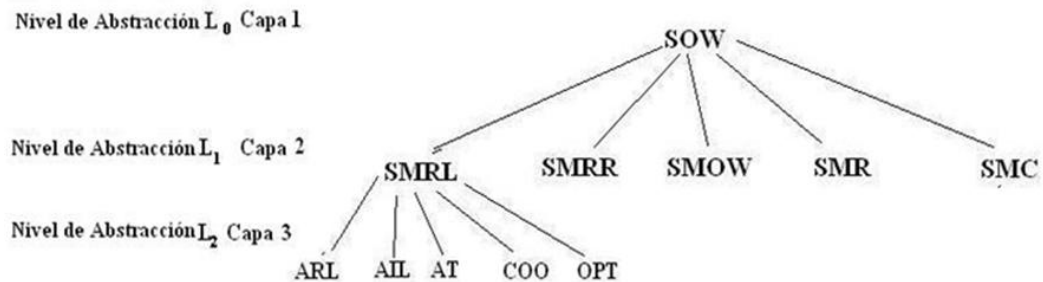


Figura 10. Niveles de Abstracción y Descomposición en el SOW

B. SOWA Nivel Micro

En éste Nivel, las plantillas de MASINA se cruzan para su verificación. Para el caso de estudio solo se presenta el cruce entre el Modelo de Agentes y el Modelo de Tareas del Agente Coordinador (COO en la Figura 10), el cual es uno de los Componentes del SMRL, que se encuentra en el Nivel de Abstracción L_2 . Las plantillas de MASINA de ese agente, y de su tarea “Recibir Solicitud de Servicio”, son las siguientes:

Tabla 1. Modelo del Agente Coordinador (Plantilla Características del Agente)

NOMBRE AGENTE	Coordinador
TIPO	Agente Software: Agente Coordinación
PAPEL	Interacción entre el SMRL y el SMR/SMRR/SMC para procesar la búsqueda de un servicio.
POSICIÓN	Es un Agente de bajo nivel dentro del SMRL
DESCRIPCIÓN	Es el Agente responsable de recibir una petición de búsqueda de servicios, clasificarla en subpeticiones y distribuirla entre el Agente ARL y el Agente AIL para su procesamiento. Finalmente, recibe las respuestas de estos agentes y emite los resultados obtenidos al subsistema solicitante.

Tabla 2. Modelo del Agente Coordinador (Plantilla Objetivo)

NOMBRE OBJETIVO	Manejar peticiones de búsqueda de servicios
TIPO	Objetivo persistente
CONDICION DE ACTIVACION	Recibir una solicitud de búsqueda de servicio de algunos de los subsistemas (SMR/SMRR/SMC)
CONDICION DE FINALIZACION	Recibir respuestas del agente ARL y/o agente AIL para integrar la respuesta a la solicitud de búsqueda de servicio.
CONDICION DE ÉXITO	Que se encuentre el servicio solicitado y este disponible
CONDICION DE FRACASO	\neg Condición de éxito
DESCRIPCION	El coordinador tiene como objetivo manejar las peticiones de búsqueda de servicios, su descomposición de subpeticiones y la distribución de estas a los agentes ARL y/o AIL. Posteriormente, en base a los resultados obtenidos emite una respuesta al subsistema solicitante.

Tabla 3. Modelo del Agente Coordinador (Plantilla Servicio)

NOMBRE SERVICIO	Armar proceso de Búsqueda
TIPO	Gratuito, concurrente
PARAMETROS DE ENTRADA	Petición de búsqueda de servicios
PARAMETROS DE SALIDA	Respuesta a la solicitud de búsqueda de servicio
LENGUAJES DE REPRESENTACION	Lenguaje Natural
ONTOLOGÍA	Ontología de servicio del SMRL

Tabla 4. Modelo de Tarea: Recibir Solicitud de Servicio

OBJETIVO TAREA	Descomponer las peticiones de búsqueda de Servicio del SMRL/SMRR/SMC
DESCRIPCIÓN	A través de ésta Tarea el sistema recibe la información sobre el servicio solicitado por el subsistema solicitante, esta información es descompuesta en subpeticiones para luego ser clasificada.
PRECONDICIÓN	La petición de búsqueda debe estar completa
FRECUENCIA	Relativa a la entrada

C. Instanciación de las Plantillas de MASINA en Protégé-OWL

Dicho ejemplo se procedió a instanciarse en nuestro marco “Ontológico de Validación de Agentes diseñados con MASINA”, desarrollado en Protégé-OWL. Para ello, Protégé-OWL procede a diseñar formas automáticamente basadas en los criterios de verificaciones que presentamos en la sección III, e introducidos en Protégé en la sección IV.

1) Nivel Macro

Las formas que se crean en el panel permiten instanciar las diferentes clases que componen el sistema, y a través de ellas describir el caso de estudio (verificar los criterios de verificación composicional). Así, para la Subclase Capa perteneciente a la Clase `A_Nivel_Macro`, las formas presentadas son: `has_componente`, `has_objetivos_macro`, `pertenece_al_Nivel`, entre otras, las cuales corresponden a las propiedades de la Clase Capa, que a su vez describen los criterios de verificación de ese nivel (Ver Figuras 7, 11 y sección 4.2).

2) Nivel Micro

Desde el punto de vista del cruce de las plantillas de MASINA, la instanciación se hace de la misma manera que a nivel Macro, es decir, a partir de las formas que se crean en el panel derivadas de los criterios de verificación para el caso de cruce de modelos hechos en la sección IV.B. Veamos el caso del cruce entre el modelo de Agentes con el Modelo de Tareas visto en la sección IV.B. La Figura 12 muestra esa instanciación de los criterios de verificación de ese cruce (ver sección III.B), usando la información de las tablas de MASINA de esos modelos mostrada en la sección V.B.



Figura 11. Instancias tomadas del Modelo de Agentes de la Clase Capa

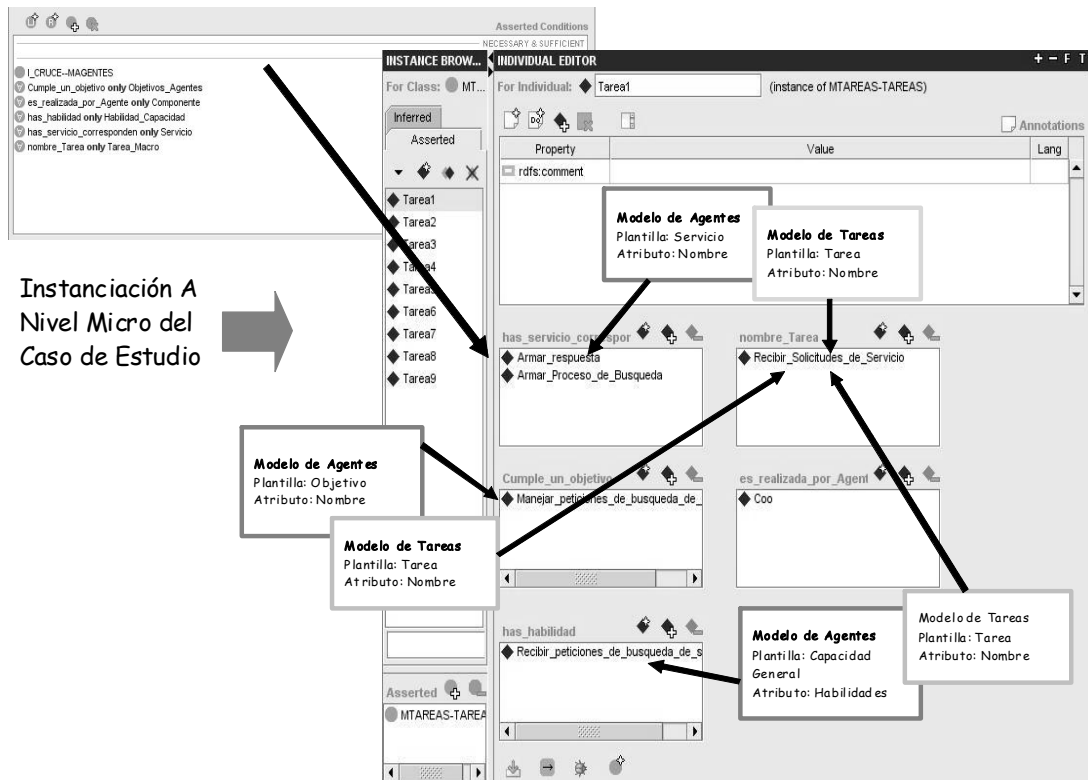


Figura 12. Instanciación de los criterios de verificación del cruce de los Modelos de Agente y tareas para nuestro caso de estudio

VI. PRUEBAS REALIZADAS Y ANÁLISIS DE LOS RESULTADOS

A. Validador básico de OWL

Para la validación de la sintaxis de las Ontologías creadas con Protegé existe un servicio que permite hacerlo. Este

Validador de Lenguajes OWL de una Ontología esta desarrollado dentro del proyecto WonderWeb [7]. La Ontología creada anteriormente se validó a través de éste servicio, dando como resultado un reporte que corrobora y valida que el lenguaje desarrollado para la Ontología es el OWL DL, y además, que sintácticamente no presenta errores (Ver Figura 13).

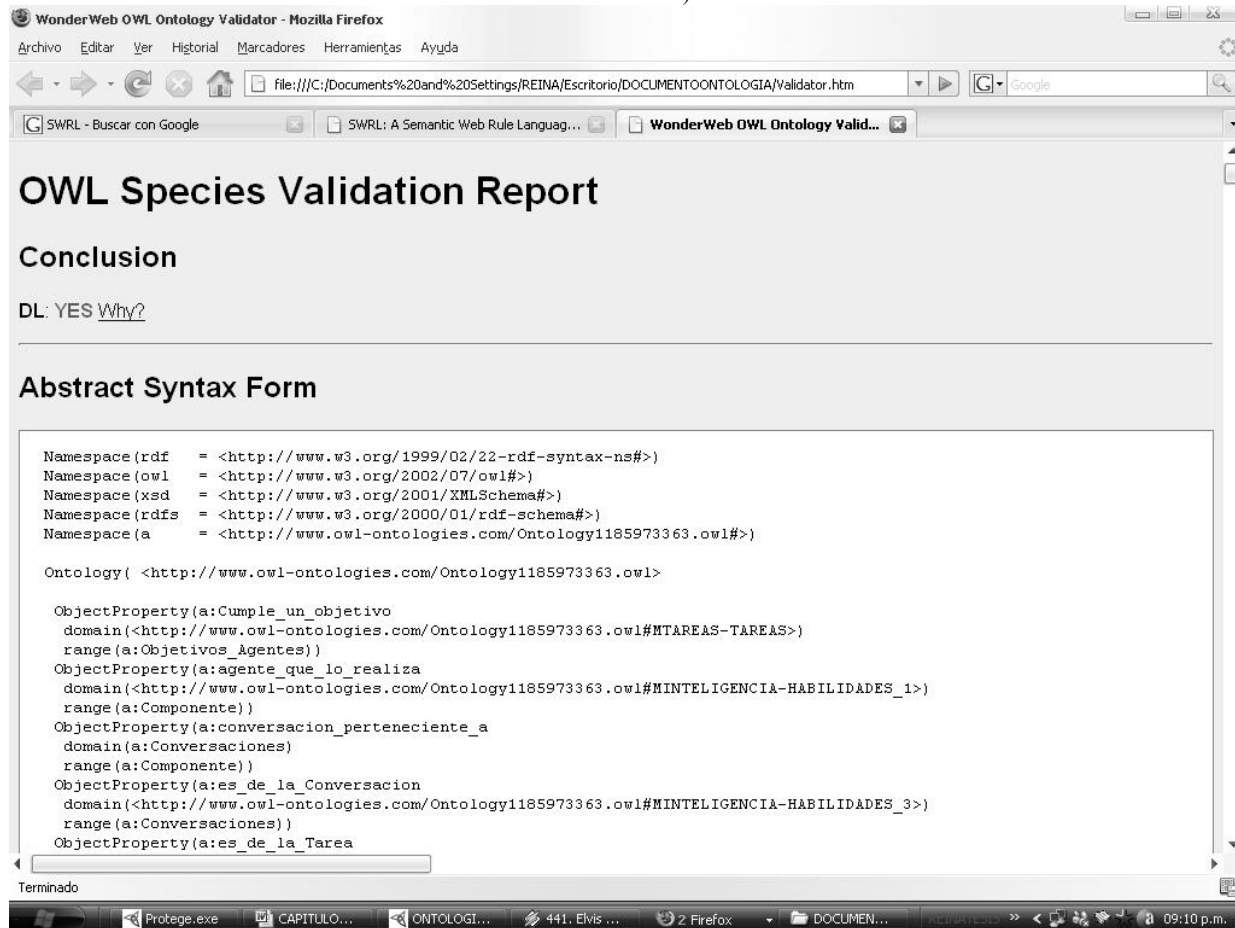


Figura 13. Validador básico de OWL

B. Análisis de la Ontología SMA aplicando Reglas Lógicas a través de SWRL

Una Ontología en OWL contiene una secuencia de hechos y axiomas. Los axiomas pueden ser de varias clases, por ejemplo, axiomas de subclases, de equivalencias de clases, restricciones, etc., como las creadas en la sección III[14]. SWRL propone extenderlos con axiomas de reglas. Un axioma de regla consiste en un antecedente y un consecuente, cada uno de los cuales esta compuesto de átomos.

Con el objetivo de garantizar el correcto funcionamiento de la Ontología desarrollada se procedió a la creación de una Regla sencilla a través de SWRL, de forma que permitan expresar restricciones más complejas. Para el caso de Estudio estudiado se estableció la siguiente regla:

Si un componente esta compuesto por otro y ambos tienen el objetivo Manejar_trazas existentes entonces el primer componente pertenece al Nivel L_0 y el segundo componente al Nivel L_1 .

En SWRL se escribe de la siguiente manera:

```

esta_compuesto(?x, ?y) ““ has_objetivos_comp(?x,
Manejar_trazas_existentes) ““ has_objetivos_comp(?y,
Manejar_trazas_existentes) ! pertenece_a_un_Nivel(?x, L0)
““ pertenece_a_un_Nivel(?y, L1)

```

Esta regla debe verificar el criterio de la sección III.A: Un Componente de la Capa L_{i+1} satisface uno o varios objetivos de la Capa L_i . Una vez creada dicha regla, se procedió a correr el motor de inferencia que SWRL utiliza llamado Jess. De

esta manera obtenemos los siguientes resultados:

- De los antecedentes de la regla se concluye que la Ontología reconoce que el componente SOW-PRIMITIVO esta compuesto por el SMRL, y además reconoce que ambos comparten el mismo objetivo.
- Al reconocerse lo anterior, y a través del motor de inferencias Jess, se instancian nuevamente los valores que intencionalmente se habían eliminado, los cuales indican

que el SOW-PRIMITIVO esta compuesto por el SMRL, es decir que el SOW-PRIMITIVO pertenece al Nivel de Abstracción L_0 y el SMRL al nivel de Abstracción L_1 .

- Con esto se verifica el cumplimiento del criterio de la sección III.A: Un Componente de la Capa L_{i+1} satisface uno o varios objetivos de la Capa L_i .

En la Figura 14 se puede observar el resultado de la Verificación del Criterio de la sección III.A.

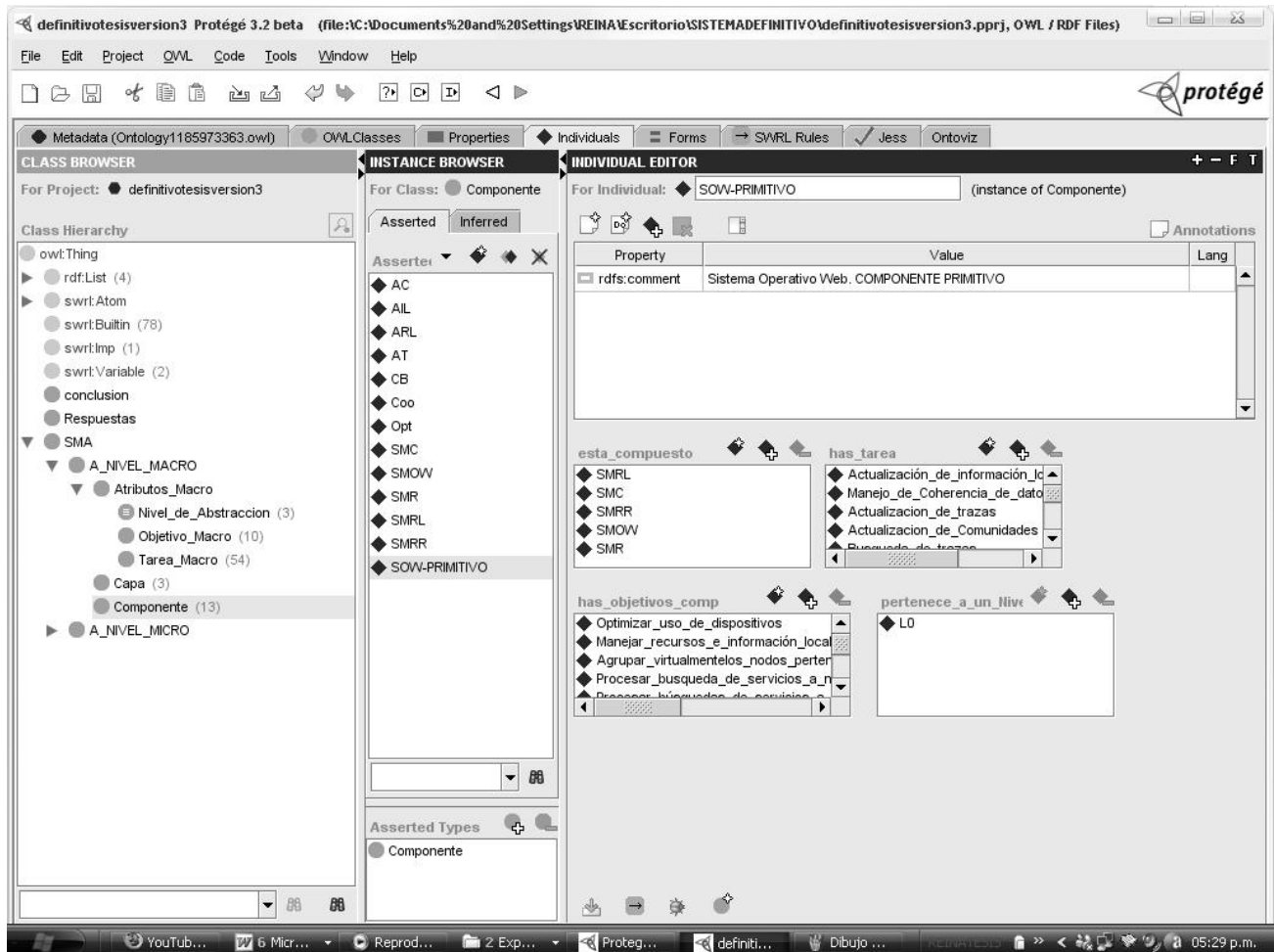


Figura 14. Resultados de la Verificación del Criterio de la sección III.A

VII. CONCLUSIONES

En éste trabajo se desarrollo un modelo ontológico para la Verificación de Agentes diseñados con MASINA. Se especificó la Ontología basada en dos técnicas de verificación: descomposición funcional de un SMA y cruce de modelos. Para poder usar las dos técnicas, la ontológica divide el proceso de verificación en dos niveles: Nivel Macro y Nivel Micro. En el nivel macro se realizan todas las verificaciones generales entre los componentes del SMA usando la verificación composicional, y en el micro se verifica la especificación de los agentes usando el cruce de modelos. A través de la Ontología

desarrollada en OWL se logró la especificación de 21 Criterios que permiten verificar el diseño de un SMA.

Esta Ontología fue implementada en Protégé-OWL, lográndose un manejo óptimo de información y de validación de la ontología usando los servicios que dicha herramienta provee. Particularmente, ella permite manipular las clases y sus propiedades (conceptos que describen el proceso de verificación), restricciones (criterios de verificación), e instancias (para usarla en casos de estudio de especificación de SMA). En nuestro trabajo se muestra el uso concreto de dicha implementación en un caso de estudio.

REFERENCIAS

- [1] Aguilar J., Cerrada M. and Hidrobo F., 2007. A Methodology to Specify Multiagent Systems, Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 4496, pp. 92-101.
- [2] Iglesias C., 1998. Definición de una metodología para el desarrollo de Sistemas Multi-Agente. Tesis Doctoral. Departamento de ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid.
- [3] Tansley D. y Hayball, C., 1993. Knowledge Based systems Analysis and Design a KADS developer's handbook. Prentice Hall.
- [4] Abadi M. y Lamport L., 1993. Composing Specifications, ACM Transactions on Programming Languages and Systems, Vol. 15, No.1, pp.73-132.
- [5] Engelfriet J. y Jonker C., Compositional Verification of Multi-Agent Systems in Temporal Multi-Epistemic Logic. <http://www.cs.vu.nl>
- [6] Aguilar J., Vizcarrondo J., Perozo N. y Ferrer F., 2005. Architecture of a Web Operating System based on Multiagent Systems, Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 3681, pp. 700-706.
- [7] <http://phoebus.cs.man.ac.uk:9999/OWL/VALIDATOR>.
- [8] Castellano, R. y Aguilar, J., 2008. Verificación de los agentes diseñados con MASINA basado en lógica de primer orden, Reporte Técnico CEMISID-10-2008, Departamento de Computación, Universidad de los Andes.
- [9] Aguilar J., Vizcarrondo J. y Perozo, N., 2006. Definition of a Verification Method for the MASINA Methodology, International Journal of Information Technology, Springer-Verlag, Vol. 12, N.3, pp. 121-131.
- [10] Jonker, C. y Treur J., 1997. Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. COMPOS 97.
- [11] Fisher, M., and Wooldridge, M., 1997. On the Formal Specification and Verification of Multi-Agent Systems. International Journal of Cooperative Information Systems, Vol. 6, pp. 67-94.
- [12] Kinny, D., Georgeff, M.P., and Rao, A.S., 1996. A Methodology and Technique for Systems of BDI Agents. Lecture Notes in Artificial Intelligence, Vol. 1038, Springer Verlag, pp. 56-71.
- [13] Frances, B. y Cornelissen, F., Compositional Design and Verification of a Multi-Agent System for One-to-Many Negotiation. <http://www.sikt.hk-r.se>
- [14] www.protege.org
- [15] Aguilar, J., Cerrada, M., Mousalli, G., Rivas, F. and Hidrobo, F., 2005. A Multiagent Model for Intelligent Distributed Control Systems, Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 3681, pp. 191-197.
- [16] Aguilar, J., Besembel, I., Cerrada, M., Hidrobo, F. y Narciso, F., 2008. Una Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes, Revista Iberoamericana de Inteligencia Artificial, AEPIA (<http://erevista.aepia.org>), Vol. 12, N. 38, pp. 39-60.