

Descubrimiento de Recursos en un Entorno Grid mediante Ontologías

Martín Chuburu* Pablo Fillottrani Javier Echaiz Jorge Ardenghi

Departamento de Ciencias e Ingeniería de la Computación
Teléfono: +54 291 4595135, Fax: +54 291 4595136
Universidad Nacional del Sur, Bahía Blanca (8000), Argentina
{mic,prf,je,jrap}@cs.uns.edu.ar

Abstract

Grid technologies allow sharing of heterogeneous resources spreaded on the Internet or a Wide Area Network. The infrastructure determined by the use of these technologies (grid computing) is only a part of a wider scenario, which also includes information management and knowledge processing support of distributed processes. This wider view is adopted by the *Semantic Grid* [6] which is defined as an extension of Grid Computing where the information and services are described in a well-defined form, allowing people and computers to work in a more cooperative way.

In this paper, the recent proposes in the ontologies management for grid resource semantic description are analyzed, which facilitate the location of resources when applications need it, and explore the possibilities of development of modules which make use of knowledge bases. In this way, it is possible to extend existing grid architectures with the aim of making the task of web resource discovery more efficient and compose this resources in an intelligent way.

Keywords: Resource Discovery, Grid Computing, Semantic Grid, Ontologies.

Resumen

Las tecnologías Grid permiten el uso compartido de recursos heterogéneos que se encuentran diseminados geográficamente en una WAN o en Internet. La infraestructura determinada por el uso de estas tecnologías (computación grid) es solamente una parte de un escenario más amplio en el que también se incluye el manejo de información y soporte para el procesamiento de conocimiento por parte de los procesos distribuidos. Esta visión más amplia es adoptada por la *grid semántica* [6] que se describe como una extensión de la computación grid donde la información y los servicios son descriptos de forma bien definida, permitiendo trabajar a las personas y a las computadoras de una manera más cooperativa.

En este artículo se analizan las recientes propuestas en el manejo de ontologías para describir de forma semántica los recursos que forman parte del entorno grid, lo cuál facilita la localización de los mismos cuando las aplicaciones necesitan de ellos, y se exploran las posibilidades de desarrollo de módulos que hagan uso de bases de conocimiento. De esta forma, es posible extender arquitecturas grid existentes con el fin de hacer más eficiente la tarea de descubrir recursos en la web y componerlos en forma inteligente.

Palabras clave: Descubrimiento de Recursos, Computación Grid, Grid Semántica, Ontologías.

1. Introducción

La localización de un servicio o un dispositivo en sistemas distribuidos de gran escala es una de las tareas más cruciales del mismo. Existe una variedad de sistemas de descubrimiento de servicios que permiten a las aplicaciones encontrar los servicios que son necesarios para llevar a cabo su tarea. Entre los sistemas tradicionales, algunos se basan en un descubrimiento a partir de comparación de nombres, mientras que otros lo hacen basados en los atributos del servicio. Estos atributos suelen pertenecer a tipos primitivos de datos, como enteros o cadenas de caracteres, por lo que el proceso de descubrimiento de servicios es realizado principalmente por vinculación de tipos (*type matching*) basado en comparación de cadenas o de enteros [12]; esto provocará que mucha información irrelevante sea devuelta al usuario si se utilizan muchas

*Becario de la *Comisión de Investigaciones Científicas* (CIC) de la Provincia de Buenos Aires, Argentina

palabras clave para la búsqueda (falsos positivos) o también se podría dar el caso de que información que podría ser devuelta no lo es por no corresponderse exactamente con los parámetros de la búsqueda (falsos negativos).

En un entorno grid donde los servicios varían tanto en nombre como en funcionalidades, resulta evidente la necesidad de que tanto los proveedores como los usuarios de estos servicios compartan un vocabulario para definir tanto los recursos ofrecidos como sus propiedades. Este vocabulario, es lo que se conoce como una *ontología*.

Con el uso de ontologías, se hace posible que el proceso de vinculación realice implicaciones sobre la jerarquía de subsunción (herencia) permitiendo el reconocimiento de vinculaciones semánticas más allá de las diferencias sintácticas que pudieran existir.

Para esto existen un número de herramientas y estándares que conformarán la base que permita la inclusión de mecanismos semánticos en el manejo y descubrimiento de los recursos de un entorno grid.

En un primer nivel, la representación de la información sobre los recursos (metadatos) en la web queda a cargo de RDF (*Resource Description Framework*). Es un lenguaje que provee un marco de trabajo común para expresar esta información de forma tal que pueda ser intercambiada sin ninguna pérdida. Los objetos en RDF son identificados a través de URIs (*Uniform Resource Identifier*) y son descriptos en términos de propiedades simples y valores asignados a esas propiedades. Provee la codificación para las relaciones binarias entre un sujeto y un objeto.

Luego, se encuentra *RDF Schema* (RDFS) que puede ser utilizado para definir vocabularios para las sentencias RDF. RDFS provee las facilidades necesarias para describir clases y propiedades específicas de la aplicación, e indica cómo utilizarlas. RDFS puede ser visto como un sistema de tipos para RDF.

Un nivel por encima, se encuentra OWL (*Web Ontology Language*) que es un lenguaje de marcado semántico utilizado para describir ontologías en términos de clases que representan conceptos y/o colección de individuos y propiedades. OWL va más allá que RDFS proveyendo medios para expresar relaciones entre clases como la disjunción, restricciones de cardinalidad, igualdad, etc. Existen 3 versiones de OWL: “Lite”, “DL” y “Full”. Las últimas dos proveen razonamiento computacionalmente completo.

En un entorno grid la administración de un recurso y su estado se hace a través de una interfaz denominada *grid service*. Estos servicios están construidos en base a la tecnología de *web services*, por lo que se hablará de ellos en forma indistinta [4].

OWL-S (*Web Service Ontology*) provee una ontología estándar para *web services*. Involucra tres partes: el perfil, el modelo y el *grounding*. El *perfil del servicio* presenta lo que hace el mismo, con información como las entradas, salidas, precondiciones y el efecto de la ejecución del servicio. El *modelo del servicio* describe cómo trabaja el mismo, esto es, cuales son los procesos que lo componen, como es que se ejecutan estos procesos y bajo que condiciones se ejecutan. En cuanto al *grounding* del servicio, especifica los detalles que el cliente necesita para interactuar con el servicio, como protocolos de comunicación, formato de mensajes, números de *ports*, etc.

En la sección 2 se presentan varios sistemas que utilizan gestión de semánticas y ontologías para el descubrimiento de recursos en un entorno grid. Y en la sección 3 se realiza una comparación de estos sistemas y se exponen las conclusiones a las que se llegaron a través del análisis de los mismos.

2. Trabajos relacionados

Existen varios trabajos en este área, algunos de los cuales agregan funcionalidades semánticas a arquitecturas de computación grid existentes, mientras que otros presentan nuevos modelos y arquitecturas haciendo uso de las tecnologías para la utilización de web services semánticos presentados en la sección 1.

En [12], se presenta un sistema en el que los servicios publican sus capacidades a través de “avisos” (*advertisements*) que son registrados en una entidad de vinculación. Esta es la que se encarga de vincular los pedidos que llegan solicitando servicios, con aquellos avisos que son “suficientemente similares” a los servicios solicitados devolviendo los avisos sobre los mismos al solicitante de forma que pueda elegir cuál es el que más le conviene utilizar. En principio, un aviso y un pedido son suficientemente similares cuando ambos describen el mismo servicio. Esto podría parecer muy restrictivo dado que no hay un acuerdo previo sobre como debe representarse un servicio.

Es necesario, por lo tanto, permitir que los motores de vinculación realicen vinculaciones más flexibles, de modo que se pueda reconocer cierto grado de similitud entre avisos y solicitudes con el fin de proveer una definición menos estricta de “suficientemente similares”. Sin embargo esto constituye un *tradeoff*, porque si se permite poca flexibilidad, se reduce la posibilidad de encontrar servicios que satisfagan los pedidos, lo cual minimiza la cantidad de falsos positivos, pero también incrementa la cantidad de falsos negativos. Por

otro lado, incrementando la flexibilidad en la vinculación se obtiene el efecto contrario. Esto es, reduce la cantidad de falsos negativos, pero aumenta la de falsos positivos.

Por lo tanto, sería deseable minimizar tanto los falsos positivos como los falsos negativos. Esto se logra, en este esquema, mediante la utilización de 3 etapas de filtrado en combinación con una ontología dentro de la entidad de vinculación. Los filtros son los encargados de minimizar los falsos positivos, mientras que la ontología restringe los falsos negativos.

Además se agregan 2 filtros más, al comienzo y al final de la etapa de vinculación, para completar el proceso de descubrimiento quedando de la siguiente manera:

- **Vinculación por el contexto:** se seleccionan aquellos avisos en la base de datos que pueden ser comparados con el pedido ya sea en el mismo contexto o en un contexto similar.
- **Vinculación sintáctica:** Este filtro compara el pedido con los avisos seleccionados por el filtro anterior en tres pasos que son: comparación de perfiles, vinculación por similitud y vinculación por signatura. La comparación de perfiles del pedido y los avisos usa una representación por palabras clave para las especificaciones. La ontología del servicio provee el modelo del servicio y el service grounding. Los últimos dos pasos se enfocan en las restricciones de entrada/salida y las declaraciones de la especificación.
- **Vinculación semántica:** Este filtro final controla las restricciones de entrada/salida de todo par pedido-aviso que fue vinculado de forma lógica.

El trabajo realizado en [16] presenta otro sistema en el que se agrega una capa de manejo de conocimiento sobre una arquitectura grid existente con el fin de poder utilizar ontologías para la descripción de los servicios disponibles y que esta se maneje de manera transparente para el sistema ya implementado que se encuentra por debajo. Para esto se utiliza una modificación sobre la arquitectura de capas para un grid propuesta por Foster en [5], quedando la siguiente jerarquía de capas:

- **Capa fábrica:** Esta es la capa de más bajo nivel, que provee los recursos para los cuales su acceso es mediado por protocolos de grid. Los recursos pueden ser recursos de procesamiento, sistemas de almacenamiento, catálogos, recursos de red y sensores o pueden ser entidades lógicas como un sistema de archivos distribuido, un cluster o un pool distribuido de computadoras.
- **Capa de núcleo de *middleware*:** Consiste de *middleware* de bajo nivel que provee un acceso seguro y unificado a recursos remotos. Dependiendo del tipo de recursos se pueden utilizar diferentes middlewares como *Globus*, *Unicore*, *Alchemi*, *SRB*. Sobre estos servicios básicos se pueden construir servicios de más alto nivel que soporten la rápida creación y despliegue de aplicaciones sobre grids.
- **Capa *middleware* de alto nivel:** Generalmente constituida por un *broker* de recursos que utiliza los servicios semánticos de descubrimiento ofrecidos por la capa de conocimiento. Los *brokers* tienen que estar diseñados para operar con diferentes frameworks de *middleware* grid y toolkits como *Globus* que corre sobre máquinas con sistemas operativos tipo UNIX, y *Alchemi* que es una plataforma de computación grid basada en .NET para sistemas Microsoft Windows. Como se puede observar, la capacidad de manejar distintos middlewares y toolkits posibilita la creación de un grid independiente de plataformas.
- **Capa de conocimiento:** Provee descubrimiento semántico a partir de una gran cantidad de datos recolectados por los de servicios de información que se encuentran por debajo (capa núcleo). Esta capa utiliza una ontología perteneciente al dominio.
- **Capa de aplicación:** Permite el uso de los recursos en un entorno grid a través de los protocolos de colaboración y acceso implementados por capas inferiores. Esta capa cuenta con un módulo de acceso semántico (*semantic portlet*) que permite al solicitante del recurso enviar la consulta y obtener el recurso adecuado usando un razonador.

Los módulos implementados en la capa de conocimiento utilizan el enfoque de web semántica para hacer la información entendible para las computadoras, esto es, términos utilizando semánticas que describan el significado de la información. Este conjunto de términos con significado son lo que definimos anteriormente como una ontología. Es utilizada para capturar conocimiento sobre cierto dominio de interés y describe los conceptos del dominio así como las relaciones que se mantienen entre estos conceptos.

Para definir las ontologías se utiliza OWL (*Web Ontology Language*) de la W3C (*World Wide Web Consortium*). OWL es desarrollado como una extensión de vocabulario de RDF y es derivado de DAML+OIL que provee mayor poder descriptivo que RDF, como el manejo de disjunciones y cardinalidad.

Protégé es un editor de OWL desarrollado por la Universidad de Standford que facilita la creación de ontologías y la construcción de bases de conocimiento. Se puede utilizar un motor de inferencias para consultar la base de conocimiento y obtener la información. En [16] se utilizó el motor de inferencia Algebron para interactuar con la base de conocimiento *Protégé* para la recuperación de información semántica.

Los dos principales módulos que constituyen la capa de conocimiento de esta jerarquía son el *módulo de descripción* y el *módulo de descubrimiento*. Ambos módulos se pueden observar en la figura 1. A continuación hablaremos un poco de cada uno de ellos.

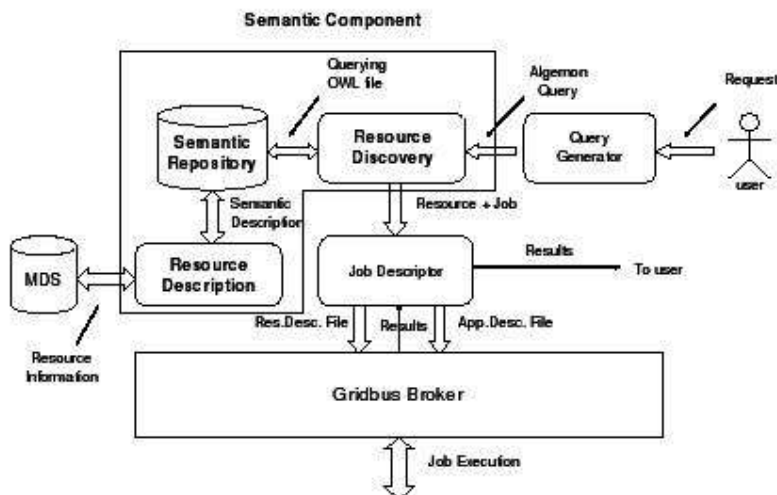


Figura 1: Capa de conocimiento

El proceso de creación de una ontología es, en principio, un proceso totalmente manual. En un entorno como es el de la computación grid, con muchos tipos de recursos ingresando y dejando el sistema constantemente es evidente la necesidad de un mecanismo que maneje de forma dinámica el mantenimiento de la ontología. Para esto se podría establecer una ontología predefinida (*ontology template*) que defina los conceptos necesarios para caracterizar los distintos tipos de recursos de forma genérica, y utilizarla en combinación con una base de conocimiento de instancias que satisfagan esa ontología representando a los recursos en sí.

El *módulo de descripción* define esta ontología predefinida utilizando el editor *Protégé* proveyendo los conceptos y propiedades necesarios con los que un recurso puede ser descrito. Cuando un recurso está registrado en el grid, su información puede ser obtenida por medio de una herramienta de monitoreo de recursos grid como puede ser *MDS* del *Globus Toolkit* [14]. Con esta información, se puede crear una instancia de un concepto apropiado representando un recurso en la ontología predefinida para cada recurso de cómputo que se encuentre en el grid. Las características de los recursos (por ejemplo, memoria libre) se definen también como propiedades del concepto apropiado en la ontología que se seleccionó para representar el recurso. Con estas características, la información sobre los recursos de todo el grid puede ser descripta de forma semántica, lo que a su vez habilita el descubrimiento semántico de recursos grid.

El *módulo de descubrimiento* permite que los usuarios consigan información sobre los recursos que necesita para ejecutar un trabajo. Este genera una consulta *Algebron* apropiada dependiendo de los requerimientos especificados por el usuario y ejecuta esa consulta sobre la base de conocimiento de la ontología para obtener los recursos que más se aproximan al pedido. El módulo trata de obtener aquellos recursos que coinciden exactamente con el solicitado. También obtiene aquellos recursos en los que se observa una relación de subsunción (herencia o jerarquía) cuando no se encuentra una coincidencia exacta, por ejemplo cuando se necesita una computadora (el recurso) con sistema operativo tipo UNIX, si se cuenta con una máquina con GNU/Linux se podría entregar este recurso ya que hay una relación de herencia. Una vez que se localiza el recurso adecuado, se crean descriptores de aplicación y de recurso con la información sobre el trabajo a realizar y el recurso obtenido. Estos descriptores son necesarios para que el *broker* pueda ejecutar la tarea en el recurso adecuado. El módulo de descubrimiento recolecta los resultados generados por el *broker* como consecuencia de la ejecución de la tarea, y pasa estos resultados al usuario que solicitó la tarea.

Otra herramienta que permite el manejo de ontologías en Grid es la presentada en [2]. Una herramienta llamada *WSDL2OWL-S* permite la generación de descripciones OWL-S para servicios con y sin estado (Grid y Web Services, respectivamente) a partir de la descripción WSDL del servicio en cuestión. El proceso de traducción comienza tomando la configuración y el URL del documento WSDL. El traductor parsea

el documento WSDL y extrae las operaciones, *port-types*, entradas, salidas y las propiedades del recurso. Luego, para cada operación WSDL se genera la sección correspondiente que finalmente formará parte del documento OWL-S. Por último se toman las entradas, las salidas, las precondiciones y los efectos y se mapea los elementos a conceptos ontológicos definidos en la configuración. Si es necesario, se creará una ontología que modele las propiedades de recurso de los servicios dados. Esta herramienta ha sido utilizada en la arquitectura K-WfGrid (*The Knowledge-based Workflow System for Grid Applications*) [1], un diseño de servicios grid semánticos destinados a simulaciones para la predicción de inundaciones.

En [10] se presenta un sistema para la atención distribuida de consultas RDF por encima de *Atlas*, una red *peer-to-peer* de proveedores y usuarios de recursos Grid. Según los autores, este enfoque va más allá del servicio de información provisto por MDS de Globus Toolkit de dos maneras significantes:

- Ofrece modelos de datos y lenguajes de consultas expresivos basados en semánticas para proveedores y solicitantes de servicios (esto es, RDF(S) y RQL en lugar de XML y XPath).
- Se implementa descubrimiento de recursos usando técnicas de sistemas *peer-to-peer* logrando distribución total, alto desempeño, escalabilidad, resistencia a las fallas, robustez y adaptabilidad.

Esto representa una tendencia que existe desde hace un tiempo atrás de utilizar técnicas del área de sistemas *peer-to-peer* en sistemas de computación grid para descentralizar y así lograr escalabilidad en aquellos componentes del sistema que aún poseen una organización jerárquica o centralizada (MDS en Globus es un ejemplo de ello).

En este artículo también se muestra como el sistema *Atlas* puede ser utilizado para dar soporte a los servicios de anotación (registro) de recursos y el servicio de ontologías. El artículo presenta como ejemplo un proveedor que quiere registrar un servicio para reparación de autos, por lo cuál primero necesita obtener una ontología sobre autos para tal fin. Cuando el servicio de anotación desea registrar (*annotate*) el servicio, primero consulta a través de RQL al servicio de metadatos provisto por *Atlas* para obtener información sobre la localización y descripción de la ontología deseada (en este caso, una sobre autos). Después de obtener esta información sobre la ontología, el servicio de anotación puede obtener la misma del servicio de ontologías.

Usando esta ontología, el servicio de anotación puede completar el proceso de registración del servicio. El resultado es almacenado en *Atlas* para cuando un usuario solicite el servicio. En la figura 2 se puede observar el proceso descrito y una posterior solicitud del servicio por parte de un cliente.

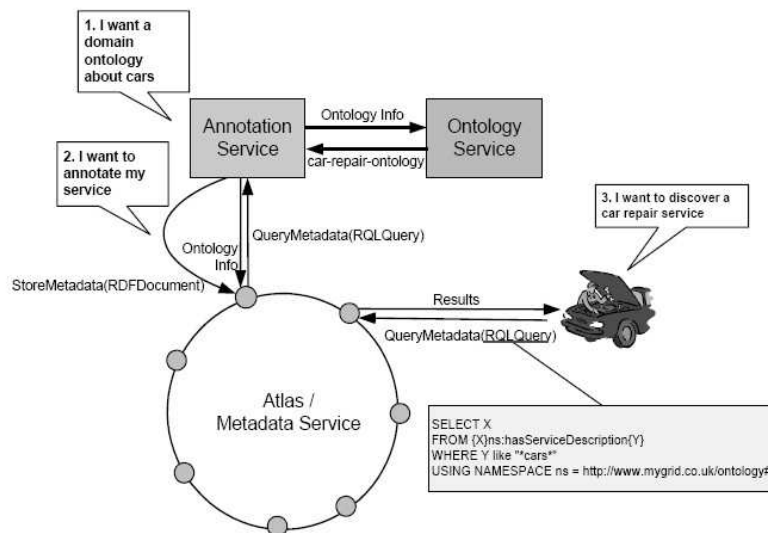


Figura 2: Utilización de Atlas para la anotación de servicios y descubrimiento

Otro trabajo en el área de manejo de semánticas en descubrimiento de recursos grid es el presentado en [13]. En el mismo, se expone una arquitectura compuesta por agentes que se encargan de realizar la búsqueda de concordancia (*matchmaking*) entre una solicitud de servicio y los proveedores de los recursos que pueden satisfacer la solicitud.

Un proveedor de servicio especifica las URLs (*Uniform Resource Locator*) de sus servicios a un agente de provisión de servicio (*Service Provision Agent*, SPA) que se registra a si mismo (si aún no lo ha hecho) en

el facilitador de directorio (*Directory Facilitator*, DF). Un usuario que quiere seleccionar un servicio realiza los siguientes pasos:

1. Realiza una instanciación de un agente de selección de servicio (*Service Selection Agent*, SSA).
2. Obtiene la lista de los proveedores disponibles (esto es, los agentes de provisión SPA) a través del facilitador de directorio (DF).
3. Se busca por un servicio que concuerde a través de tres pasos:
 - a) Se envía la descripción del servicio solicitado como un archivo OWL-S a los proveedores disponibles obtenidos en el paso 1.
 - b) Del lado de los proveedores, cada agente SPA calcula posibles concordancias (*matches*) en paralelo.
 - c) Los agentes SPA envían de forma asíncrona sus resultados al agente SSA solicitante.
4. Se presentan los resultados al usuario que hizo la solicitud.

En la figura 3 se presenta un gráfico UML con el proceso descrito anteriormente.

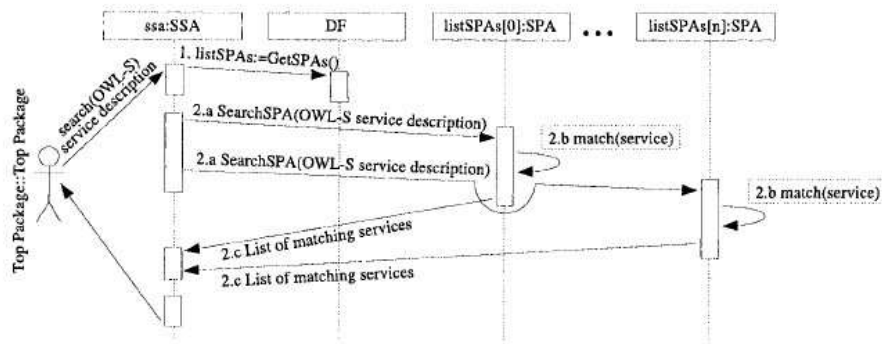


Figura 3: Seleccionando servicios

3. Conclusiones y trabajos futuros

A lo largo de este artículo se han explorado las características de los sistemas de descubrimiento de recursos actuales con los que cuentan los sistemas de computación grid y cómo estos pueden ver mejorado su desempeño a través de la utilización de herramientas para la descripción semántica de recursos y servicios web.

También se han explorado un número de propuestas que hacen uso de estas tecnologías para el mejoramiento de sus sistemas. La mayoría de las mismas exponen un diseño en el que se provee un módulo o capa para el manejo de ontologías y generación de descripciones semánticas para los recursos, de forma de actuar como interface entre los clientes y la capa de *middleware* de grid que generalmente hacen un manejo sintáctico de la información sobre los recursos.

La propuesta mostrada en [12], sin embargo, está más enfocada en el proceso de vinculación entre las solicitudes de los usuarios y los recursos que se encuentran disponibles para su utilización. Este proceso de vinculación se sustenta en la utilización de dos elementos clave. En primer lugar, una serie de etapas de filtros a través de las cuales va consiguiendo el conjunto de recursos que posiblemente puedan satisfacer los requerimientos de los usuarios. El objetivo de estas etapas es filtrar aquellos recursos que por un contexto similar a la solicitud son elegidos, sin embargo no son adecuados para satisfacer la solicitud (lo que en la sección 1 se identificó como *falsos positivos*). Complementariamente, se cuenta con una Ontología destinada a proveer un soporte para armonizar las diferencias existentes en los conceptos que determinan las propiedades y atributos de los recursos de forma que estas no constituyan una limitante al momento de asociar un pedido con un recurso, evitando así lo que denominamos anteriormente como *falsos negativos*. Si bien, el proceso de vinculación es fundamental para el descubrimiento semántico de recursos, este artículo deja desprotegido el aspecto de la descripción semántica de los mismos, y de cómo esta es obtenida a partir de la descripción sintáctica utilizada por los sistemas convencionales de computación grid.

En este sentido, en [16] se mostraba una clara separación entre la descripción de los recursos y su descubrimiento. En este enfoque se propone una capa de conocimiento por encima de la de *broker* encargada de la administración de los recursos físicos del grid. Sin embargo tiene la desventaja de ser una propuesta basada en el broker *Gridbus* [3] que es un *middleware* propietario y además, esta capa de conocimiento está integrada al mismo. Un enfoque alternativo debería ser totalmente independiente del *middleware* de grid y de cualquier servicio de información específico.

Otra desventaja menos importante de este enfoque, es la utilización del motor de inferencia *Algernon* [8] que se ha visto superado por motores más modernos como *Pellet OWL Reasoner* [15] que es el motor utilizado por la propuesta mostrada en [13]. Este razonador de OWL es de código abierto y muy utilizado por su buena compatibilidad con Java y su buen desempeño para ontologías de pequeñas a moderadas.

Otros razonadores que también se podrían utilizar son FaCT (*Fast Classification of Terminologies*) [9] y Racer [7].

Siguiendo con los trabajos revisados, en [10] se propone un sistema de información Grid basado en DHT. Este tipo de trabajos se encuentra en un área de intersección entre Computación Grid y Sistemas *peer-to-peer*. En el mismo se afirma que la propuesta va más allá de las capacidades de MDS debido a que se basa en un modelo de datos expresivo basado en semánticas y un sistema de información descentralizado basado en P2P. Sin embargo, no se muestra en ningún momento el modelo de información semántico utilizado [11]. La propuesta hace una suposición sobre la existencia de RDF sobre el modelo de datos semántico, sin embargo este modelo es un componente de bastante importancia como para dejarlo como una suposición. Además no queda claro como se logra el mantenimiento del estado dinámico de los recursos usando un sistema de información P2P, aspecto que ya estaba cubierto por MDS.

De los trabajos que se estudiaron, resultó de especial interés la propuesta mostrada por [16] ya que hace utilización del sistema MDS del *Globus Toolkit*, ya que algunos de los autores del presente artículo se encuentran realizando investigación en computación grid utilizando este *middleware* como soporte, por lo que podrían tomarse estas experiencias para la implementación de algún componente que mejore el desempeño ofrecido por MDS a través del uso de ontologías y descripciones semánticas de los recursos grid.

Sin embargo, no sólo es importante mejorar la interfaz con el usuario agregando estas capacidades semánticas que permitan describir los recursos buscados con más flexibilidad. También es importante mejorar la capa de *middleware* a través de la utilización de técnicas *peer-to-peer* (como se mostró en [10]) que permitan descentralizar el manejo de la información sobre los recursos disponibles en un entorno grid, que es una de las falencias de los sistemas actuales de información de recursos grid.

En un próximo trabajo se considerarán estas observaciones con el fin de mejorar la utilización del *middleware* *Globus Toolkit* a través de la implementación de módulos que manejen descripciones semánticas para los recursos, y almacenamiento *peer-to-peer* de la información sintáctica sobre la cuál se mapean estas descripciones.

Referencias

- [1] Marian Babik, Emil Gatjal, Ondrej Habala, Ladislav Hluchy, Michal Laclavik, and Martin Maliska. Semantic grid services in K-Wf Grid. In *2006 International Conference on Semantics, Knowledge and Grid (SKG 2006), 1-3 November 2006, Guilin, China*, page 66. IEEE Computer Society, 2006.
- [2] Marian Babik, Ladislav Hluchy, Jacek Kitowski, and Bartosz Kryza. Generating semantic descriptions of web and grid services. In Peter Kacsuk, Thomas Fahringer, and Zsolt Nemeth, editors, *Distributed and Parallel Systems - From Cluster to Grid Computing (Proceedings of the 6th Austrian-Hungarian Workshop on Distributed and Parallel Systems (DAPSYS))*, International Series in Engineering and Computer Science (ISECS), pages 93–102, Innsbruck (Austria), 2006. Springer.
- [3] R. Buyya. The gridbus toolkit for grid and utility computing. In *CLUSTER*. IEEE Computer Society, 2003.
- [4] Martin Chuburu, Javier Echaiz, and Jorge Ardenghi. Monitoreo de Recursos computacionales en un cluster utilizando Grid Services. *IX Workshop de Investigadores en Ciencias de la Computación, WICC 2007*, pages 612–616, May 2007.
- [5] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organization. *The International Journal of High Performance Computing Applications*, 15(3):200–222, Fall 2001.

- [6] Carole A. Goble and David De Roure. The semantic grid: Myth busting and bridge building. In Ramon López de Mántaras and Lorenza Saitta, editors, *ECAI*, pages 1129–1135. IOS Press, 2004.
- [7] V. Haarslev and R. Möller. Racer: An owl reasoning agent for the semantic web. In *Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with the 2003 IEEE/WIC International Conference on Web Intelligence, Halifax, Canada, October 13*, pages 91–95, 2003.
- [8] Micheal Hewett. The Algernon Abstract Machine: Compiling and Executing Rule-Based Programs. Technical Report AI00-284, The University of Texas at Austin, Department of Computer Sciences, April 24 2000. Mon, 28 Apr 103 21:15:33 GMT.
- [9] Ian Horrocks. The FaCT system. In Harrie de Swart, editor, *Proc. of the 2nd Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 307–312. Springer, 1998.
- [10] Z. Kaoudi, I. Miliaraki, M. Magiridou, E. Liarou, S. Idreos, and M. Koubarakis. Semantic Grid Resource Discovery in Atlas. *Knowledge and Data Management in Grids*, 2006. Talia Domenico and Bilas Angelos and Dikaiakos Marios D. (editors), Springer.
- [11] Frederick Lee, Suresh Garg, and Shishir Garg. OWL-SR: Unified Semantic Service and Resource Discovery for Grids. *OWL-S: Experiences and Directions. A workshop at The 4th European Semantic Web Conference (ESWC 2007)*, June 2007.
- [12] Simone A. Ludwig and Peter Van Santen. A grid service discovery matchmaker based on ontology, December 10 2002.
- [13] Konstantin Popov, William Groleau, and Vladimir Vlassov. Towards semantics-based resource discovery for the grid. In *Proceedings of CoreGRID Integration Workshop*, Pisa, Italy, 2005. Preliminary proceedings.
- [14] Jennifer M. Schopf, Mike D'Arcy, Neill Miller, Laura Pearlman, Ian Foster, and Carl Kesselman. Monitoring and Discovery in a web services framework: Functionality and performance of the Globus toolkit's MDS4. Technical Report ANL/MCS-P1248-0405, Argonne National Laboratory, 2004.
- [15] Evren Sirin, Bijan Parsia, Bernardo C. Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.
- [16] T.S. Somasundaram, R.A. Balachandar, V. Kandasamy, R. Buyya, R. Raman, N. Mohanram, and S. Varun. Semantic-based grid resource discovery and its integration with the grid service broker. *Advanced Computing and Communications, 2006. ADCOM 2006. International Conference on*, pages 84–89, Dec. 2006.