

# Propiedades de Calidad de Servicio en el Descubrimiento de Recursos Grid

David Buján Carballal<sup>1</sup>, Oscar Corcho García<sup>2</sup>, Josuka Díaz Labrador<sup>1</sup>

<sup>1</sup> Facultad de Ingeniería - ESIDE, Universidad de Deusto,  
Avenida de las universidades 24, 48007 Bilbao, Vizcaya  
{dbujan, josuka}@eside.deusto.es

<sup>2</sup> Facultad de Informática, Universidad Politécnica de Madrid,  
Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid  
ocorcho@fi.upm.es

**Resumen.** Uno de los problemas abiertos en el contexto de las Arquitecturas Orientadas a Servicios es del descubrimiento de recursos y/o servicios adecuados para llevar a cabo una tarea determinada. Los proveedores de información Grid básicamente ofrecen información funcional sobre los recursos Grid que monitorizan, por lo que los modelos de información Grid básicamente representan esta información sintáctica, y los consumidores de información Grid usan normalmente dichas propiedades funcionales para seleccionar recursos. En la práctica, muchos trabajos se reinician debido a fallos en los recursos, aunque existen iniciativas que tratan de usar técnicas aisladas para manejar algunas propiedades de calidad de servicio. En el presente artículo se propone un nuevo enfoque para modelar recursos Grid junto con propiedades de calidad de servicio. Por un lado, este modelo está basado en una ontología desarrollada para integrar los modelos existentes tanto a nivel de representación de información Grid como de calidad de servicio en general. Por otro lado, también propone la creación de un sistema de medida - actualmente en desarrollo - para algunas propiedades de calidad de servicio (disponibilidad, rendimiento y fiabilidad).

**Palabras Clave:** Grid Semántico, Computación Grid, Descubrimiento de Recursos, Calidad de Servicio, Servicios Grid Semánticos.

## 1 Introducción

La Grid Semántica se define como una extensión de la Grid actual en la que la información y los servicios se presentan con un significado bien definido a través de descripciones que puedan ser procesadas por una máquina y cuyo objetivo es maximizar el potencial de la compartición y reutilización de los recursos descritos [1].

Tal y como se comenta en el resumen, uno de los problemas abiertos en el contexto de las Arquitecturas Orientadas a Servicios, como la Grid, es del descubrimiento de recursos adecuados para llevar a cabo una tarea determinada dentro de una aplicación [2,3].

La mayor parte de los enfoques de descubrimiento de servicios utilizados en estas arquitecturas están basados en el análisis de sus propiedades funcionales, es decir, en aquellas características que definen qué es lo que hace el servicio o qué ofrece. Sin embargo, existen otro tipo de propiedades no funcionales o de calidad de servicio que se pueden utilizar con este objetivo y sobre las cuales no se ha trabajado en profundidad aún.

En muchos casos, nos encontramos con situaciones en las que los trabajos que se envían a la Grid fallan debido a que los recursos que se les asignaron no estaban disponibles o habían caído. Ello provoca que los trabajos deban ser reenviados de nuevo y que se realice una nueva búsqueda y asignación de recursos. Además de la pérdida del trabajo realizado, también se produce un derroche en el uso de los recursos, así como un incremento de tiempos nada deseable.

Por un lado, existen herramientas Grid que permiten reanudar algunos tipos de trabajos desde el punto de fallo, pero no es el caso general y además tampoco resuelve el problema de la pérdida de trabajo y tiempo. Aún así, los sistemas tolerantes a fallos basados en la recuperación de trabajos son muy costosos en sistemas distribuidos, por lo que en muchos casos se suele optar por cancelar aquellos trabajos que no han terminado al cabo de cierto tiempo límite y volver a enviarlos de nuevo.

Por otro lado, se suele recurrir también a monitorizar los recursos que fallan y ponerlos manualmente como requisitos del trabajo que se lanza, de manera que nunca se envía el trabajo a esos “malos” recursos. También se suele especificar como requisito del trabajo el tiempo esperado para la finalización del mismo, de manera que se le pueda asignar un recurso cuyo tiempo de respuesta sea menor que dicho tiempo esperado de finalización del trabajo. Y la misma idea se suele aplicar a la hora de considerar el tamaño de los ficheros utilizados o la salida esperada por un trabajo, en relación al espacio físico disponible en los recursos de almacenamiento.

Sin embargo, todas estas soluciones son muy específicas y claramente mejorables.

Gran parte de los servicios desarrollados para el middleware de la Grid, que también podemos denominar como consumidores de información Grid, necesitan tener conocimiento sobre el comportamiento de los recursos Grid para poder realizar una adecuada selección y asignación de los mismos que permita ejecutar trabajos en sistemas tan dinámicos y complejos como los sistemas Grid. De cara a obtener estas propiedades no funcionales o de calidad de servicio sobre los recursos para construir dicho conocimiento, las herramientas que podemos denominar proveedores de información Grid tienen que seleccionar, medir y analizar distintas métricas sobre los recursos (sobre su comportamiento ejecutando trabajos, almacenando datos, etc.) para ofertarlas a los consumidores de información Grid. Sin embargo, no se ha realizado hasta la fecha un amplio estudio sobre métricas de calidad de servicio que pueda ser usado para evaluar la calidad de servicio de un recurso Grid.

Además, dada la complejidad tanto de los sistemas Grid como de los conceptos de calidad de servicio, se ha de realizar un esfuerzo por especificar semánticamente las relaciones entre los distintos tipos de recursos Grid y sus propiedades no funcionales o de calidad de servicio, así como el significado de cada uno de los términos utilizados. Existen iniciativas para describir modelos de información Grid por un lado, y conceptos de calidad de servicio por otro, pero no hay ninguna relación entre ellos.

Por un lado, los modelos de información Grid existentes no usan el mismo lenguaje para representar los mismos conceptos sobre recursos, aunque varios grupos de la

comunidad Grid han detectado la necesidad de integrar sus modelos y hacerlos interoperables. Además, existe una falta de representación de propiedades no funcionales o de calidad de servicio sobre los recursos. En el mejor de los casos, algunos modelos especifican un rango de valores para medir y comparar el rendimiento de los recursos Grid, pero faltan muchos detalles en general. Algunos proveedores y consumidores de información Grid desarrollados específicamente para algunos proyectos usan técnicas aisladas para tratar el rendimiento de los recursos.

Por otro lado, la mayoría del trabajo existente en cuanto a modelos de calidad de servicio en general se centran en flujos de trabajo y servicios Web en arquitecturas SOA, pero algunas de las métricas especificadas en estos modelos no son válidas para entornos Grid, dado que los recursos y sus características no funcionales son más diversas, dinámicas e inter-organizacionales. Esa es la razón por la que es necesario un nuevo modelo con métricas adaptadas a la Grid.

Sin embargo, el trabajo realizado en ambos tipos de modelos es un buen punto de partida para analizar las posibilidades de integración de las técnicas aisladas utilizadas en ambos campos. El objetivo que se persigue con el presente trabajo es desarrollar un nuevo modelo que sea integrable con los sistemas de información Grid y enriquezca el descubrimiento y selección de recursos en la Grid usando información semántica asociada a los mismos.

## 2 Estado del Arte

El middleware de la Grid es abundante y hay varias herramientas o aplicaciones que se pueden utilizar como proveedores de información Grid sobre los recursos que se encuentran desplegados en un sistema Grid y otras tantas aplicaciones potenciales consumidoras de dicha información.

Aunque los proveedores de información por excelencia son las herramientas de monitorización de recursos como Ganglia [4], Nagios [5], GridICE [6], Hawkeye [7], CluMon [8], INCA [9] o MonaLISA [10], existen muchas otras aplicaciones que pueden ser usadas con éste mismo propósito: Globus middleware [11] (GRAM, MDS, RFT, RLS, CAS), herramientas de monitorización ad-hoc de proyectos de investigación (GEMLCA-GMT [12], SCALEA-G [13], GRASP [14]), gestores locales de colas de trabajos (Condor-G [15], PBS/Torque [16], SGE [17], LSF [18], Maui [19]), así como cualquier servicio WSRF [20] que publique propiedades de los recursos en los sistemas de información Grid.

Actualmente, la mayoría de las herramientas o aplicaciones que realizan funciones de planificación, selección y/o gestión de recursos en la Grid realizan descubrimiento de recursos basado en propiedades funcionales, como por ejemplo características del procesador, tamaño de la memoria, espacio de almacenamiento disponible, software que se encuentra instalado, tiempo de respuesta a las peticiones de servicio, etc, ya que ésta es básicamente la información que ofrecen los proveedores de información Grid comentados anteriormente.

Las aplicaciones que podríamos considerar como consumidores de información Grid son, por ejemplo, GridWay meta-scheduler [21], GridARM Askalon's Grid Resource Management System [22], LCG/g-LITE broker [23], Nimrod-G Grid

Resource Broker [24], EUROGRID/GRIP Resource Broker [25], Storage Resource Broker [26], NGS Resource Broker [27], Moab Grid Scheduler (aka Silver) [28], EGEE Workload Manager Service (WMS) [29], GRMS (Grid(Lab) Resource Management) [30], Gridbus Grid Service Broker [31], Advanced Resource Connector (Nordugrid's ARC) [32], así como cualquier otro middleware de la Grid: WebMDS [33], interfaces de usuario Grid (portales Grid como P-GRADE [34]), herramientas Grid basadas en CSF [35], WebCom-G [36], GPT [37], middleware HPC4U [38] o cualquier cliente WSRF que busque y seleccione recursos.

Algunas de estas iniciativas, como por ejemplo GridWay y GridARM, tratan de incorporar en sus algoritmos de selección criterios de rendimiento y disponibilidad, pero utilizan enfoques aislados y ad-hoc.

## 2.1 Modelos de Información Grid

En los últimos encuentros de la comunidad Grid en el marco del Open Grid Forum (OGF) varios grupos relacionados con el modelado de datos e información de entornos Grid han planteado la necesidad de unificar criterios y desarrollos para intentar que los distintos modelos que propone cada grupo sean interoperables o incluso lleguen a converger en una misma especificación más genérica, dado que cada grupo enfoca el modelado desde sus propios casos de uso.

Existen iniciativas que, desde distintos enfoques y objetivos, tratan aspectos relacionados con la representación de información sobre los recursos y el propio entorno de la Grid: GLUE Schema (GLUE-WG) [39], EGA Reference Model (EGARM) [40], OGSA (OGSA-WG) [41], Reference Model (RM-WG) [42], OGSA Basic Execution Services (OGSA-BES-WG) [43], Job Submission Description Language (JSDL-WG) [44], Information Modelling in OGSA [45], CIM based Grid Schema (CGS-WG) [46], GOM [47], Grid Ontology [48].

Del mismo modo, hay otros grupos de trabajo del OGF cuyos objetivos también se alinean con la planificación y/o gestión de recursos, el descubrimiento de los mismos, o el planteamiento de propiedades no funcionales para la Grid y en algún caso aislado la medición de estas propiedades. En varios casos, la actividad de estos grupos [49] es prácticamente nula o poco destacable: OGSA Resource Selection Services (OGSA-RSS-WG), Scheduling and Resource Management (SRM), Grid Reliability and Robustness (GRIDREL-RG), Trusted Computing (TC-RG), Network Measurements Working Group (NM-WG), Grid Benchmarking (GB-RG), OGSA Resource Usage Service (RUS-WG), Usage Record (UR-WG), Grid Computing Environments (GCE-RG), Semantic Grid (SEM-RG), GSMO [50].

## 2.2 Modelos de Información sobre Calidad de Servicio

En el contexto de la representación y uso de la calidad de servicio en general, algunos de los trabajos encontrados describen aspectos relacionados con el descubrimiento de recursos a través de propiedades no funcionales. MOQ [51] propone un marco general y más bien filosófico para afrontar cualquier problema de concreción de aspectos cualitativos abstractos al mundo real; sin embargo, a pesar de que aporta información

sobre el dominio de calidad de servicio o QoS en relación a cómo normalizar o concretar la evaluación de aspectos cualitativos, no se especifica un modelo. Sánchez-Macián et al [52] trata de relacionar QoS con Quality of Experience (QoE) y Quality of Business (QoBiz). Tosic et al [53] enumera algunos requisitos que se deben tener en cuenta a la hora de desarrollar una ontología para modelar aspectos de QoS. FIPA-QoS [54] y WS-QoS [55] son ontologías sin implementación OWL, a diferencia de OWL-QoS [56] (anteriormente DAML-QoS), OWL-QoS de Maximilien y Singh [57] y QoSOnt [58], que sí cumplen con la especificación OWL [59]. QoSOnt es una ontología OWL para QoS que está especialmente enfocada a los servicios web y que fue desarrollada por la Universidad de Lancaster como parte del proyecto DIRC [60] y actualizada en el proyecto SeCSE [61].

### 2.3 Análisis Comparativo para los Modelos de Información

Hasta la fecha, todas las iniciativas mencionadas anteriormente relacionadas con la calidad de servicio en general poseen ciertas limitaciones que hacen que sea difícil adoptarlas en entornos Grid de producción. En muchos casos no son extensibles, por lo que no se pueden adaptar a la Grid. En otros casos, no son completas, centrándose sólo en una propiedad no funcional y aportando una solución ad-hoc a un problema o dominio de aplicación concreto. En la mayoría de los casos ni siquiera se plantea cómo se realiza la medición de las propiedades no funcionales, o esta es muy trivial y no deja de ser un factor de corrección un tanto aleatorio y con poco peso. Además, en el caso concreto de la Grid, las propuestas actuales son ad-hoc para proyectos concretos y, en general, o no están esquematizadas, o no se ha llevado a cabo un análisis detallado, o no plantean un modelo común para los recursos y sus propiedades. Hemos realizado un estudio más detallado que se puede encontrar en [62], en el que se comparan los distintos modelos de información en base a completitud del modelo (nivel de detalle en cuanto a la representación de tipos de recursos Grid y/o propiedades QoS), métricas utilizadas para la QoS, estándares y tecnologías que cumple o utiliza, y aplicaciones que soportan el modelo.

Por lo tanto, además de abordar el problema del descubrimiento y la selección de recursos en entornos Grid, el presente trabajo plantea la creación de un modelo de representación de recursos para entornos Grid en el que se puedan reflejar propiedades no funcionales de los recursos y propone un sistema de medida de dichas propiedades que luego permita a las potenciales herramientas o aplicaciones Grid consumidoras de este tipo de información ver mejorada su funcionalidad.

## 3 Nuestro Modelo

En el campo de la Ingeniería del Software, una ontología es una especificación de una conceptualización o una descripción de los conceptos y relaciones que pueden existir en un dominio de aplicación para una comunidad de agentes relacionados. Las ontologías permiten compartir y reusar conocimiento. Las aplicaciones pueden ser escritas de acuerdo a un compromiso de utilización del vocabulario definido por una ontología para un dominio concreto. De esa manera, los agentes que comparten la

misma ontología pueden comunicarse entre ellos ya que todos entienden el mismo vocabulario.

Teniendo en cuenta las especificaciones y modelos presentados por otros grupos de trabajo, proponemos una ontología para describir los tipos de recursos, servicios, roles y otros conceptos del dominio de la Grid, así como para representar algunas de las propiedades no funcionales o de QoS más importantes asociadas a los mismos según la literatura del área: rendimiento, disponibilidad y fiabilidad. Cuando diferentes herramientas proveedoras de información Grid soportan una ontología común, los usuarios Grid y las aplicaciones consumidoras de información Grid pueden beneficiarse de manejar una misma conceptualización sobre los recursos Grid y su comportamiento en relación a la calidad de servicio. Nuestro modelo, todavía en desarrollo, también plantea un sistema formal de medida para propiedades no funcionales o de calidad de servicio, cada una de las cuales tiene diferentes características (complejidad de cálculo, métricas, parámetros de relevancia, etc.).

Actualmente estamos definiendo, estudiando y asociando varias métricas con las propiedades de QoS sobre recursos Grid comentadas anteriormente, así como rangos para algunas de ellas con el objetivo de poder realizar comparación de valores. Nuestro modelo considera métricas para recursos Grid software y hardware a varios niveles: sistema, servicio, aplicación, nodo, red, máquina, etc.

Aquí sólo presentamos una visión general de la ontología que hemos desarrollado para representar recursos Grid y sus propiedades no funcionales o de QoS, así como una introducción a cómo medir la fiabilidad de los recursos Grid.

### **3.1 Ontología para Recursos Grid y sus Propiedades de Calidad de Servicio**

A la hora de desarrollar nuestra ontología hemos creado nuestras propias clases, pero también hemos intentado reusar y/o extender ciertas clases para integrar algunas ideas interesantes de las diferentes ontologías comentadas en el estado del arte: GOM Resource ontology, GOM Service ontology [47], Grid Foundational Ontology, Globus Ontology [48]. Estas ontologías han servido de punto de partida para diseñar una clasificación de recursos Grid más completa, teniendo en cuenta diferentes niveles de detalle, así como los diferentes roles que se pueden asociar a cada recurso en un entorno Grid. Dado que estas ontologías de partida poseen una pobre definición de propiedades no funcionales o de calidad de servicio, también hemos integrado algunas ideas presentes en la ontología QoSOnt [58] en nuestro modelo.

Debido a limitaciones de espacio, aquí solo podemos presentar una visión general de la ontología utilizando imágenes que representan las clases más importantes relacionadas con la clasificación de tipos de recursos Grid, roles y propiedades de calidad de servicio. En las imágenes aparecen puntas de flechas negras en algunas clases indicando que poseen subclases o superclases. Existen otras relaciones entre clases que se expresan como propiedades de las mismas y restricciones de lógica descriptiva que aquí no aparecen. La ontología está disponible en [63] para poder consultar el resto de clases, atributos y relaciones semánticas, así como para su pública revisión y comentario.

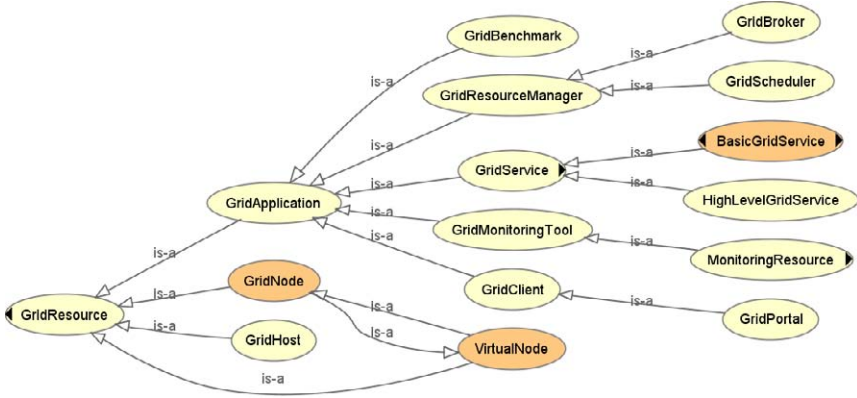


Fig. 1. Diagrama parcial de clases relacionadas con tipos de recursos Grid.

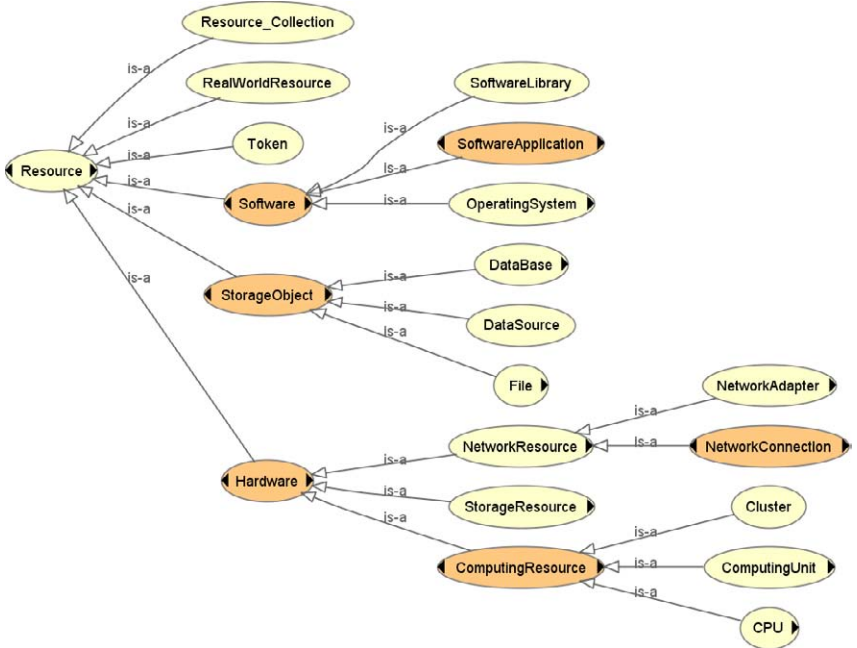


Fig. 2. Diagrama parcial de clases relacionadas con tipos de recursos en general.

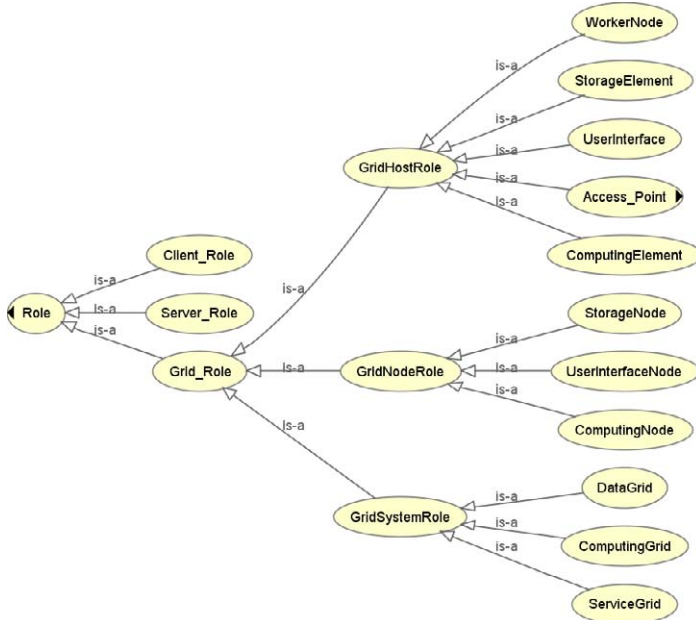


Fig. 3. Diagrama parcial de clases relacionadas con tipos de roles de los recursos Grid.

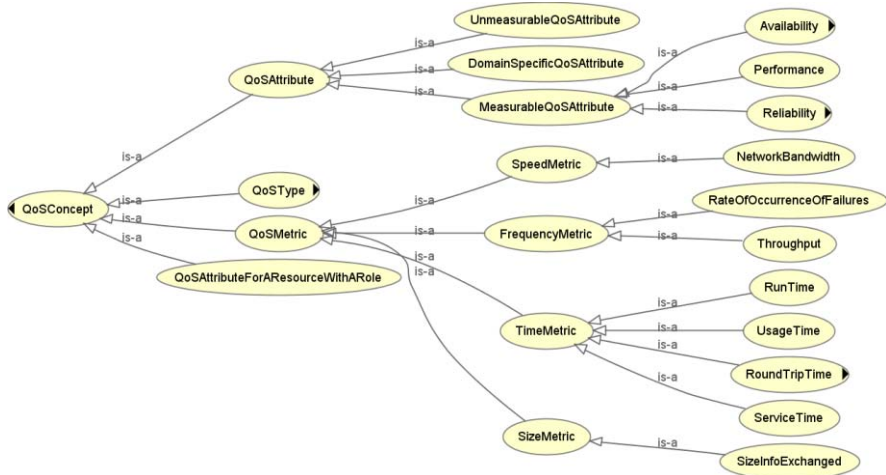


Fig. 4. Diagrama parcial de clases relacionadas con conceptos de QoS.

Dado que los lenguajes OWL no pueden manejar directamente los valores numéricos que se representan en la ontología, estos se asocian a rangos de valores para que los consumidores de información puedan realizar comparaciones de estas propiedades de los recursos. Por lo tanto, nuestro modelo tiene un enfoque híbrido, ya que se basa en una ontología para representar la información y en otras técnicas a la hora de utilizarlo en la selección de recursos.



### 3.2 Sistema de Medida para la Fiabilidad de los Recursos Grid

Según la RAE, la fiabilidad es la probabilidad de buen funcionamiento de algo [64]. Según la Wikipedia, se define como la probabilidad de que el comportamiento de un sistema o dispositivo desarrolle una determinada función, bajo ciertas condiciones y durante un período de tiempo determinado [65]. Según el IEEE la definición de fiabilidad (reliability) es la habilidad de un sistema o componente para realizar sus funciones bajo unas condiciones dadas durante un período de tiempo específico, es decir, para mejorar sus condiciones en un tiempo dado [66]. Según el Free On-line Dictionary of Computing se trata de un atributo de cualquier sistema que consistentemente produce los mismos resultados, preferiblemente alcanzando o superando sus especificaciones [67]. Como resumen de estas definiciones podemos concluir que la fiabilidad es la probabilidad de que un elemento funcione de una manera correcta y consistente, entendiendo correcta como la forma esperada y consistente como determinista, en un tiempo determinado y en un entorno dado, y se calcula así:

$$Fiabilidad = e^{-\frac{Tiempo}{MTBF}} \quad (1)$$

Para obtener el factor MTBF (Mean Time Between Failure) o tiempo medio entre fallos hay que determinar las variables necesarias para calcularlo: tamaño de la población, tiempo de procesado de la información recibida de los clientes, definir todos los posibles fallos, tiempo de diagnóstico y reparación de fallos, así como la tasa anual de fallos o AFR (Annual Failure Rate), cuya fórmula es:

$$AFR = \frac{F \times \frac{52 \text{ semanas}}{\text{año}}}{\frac{N}{P}} \quad (2)$$

donde F es el número de fallos en un mismo periodo, N el número de semanas en dicho periodo y P el tamaño de la población de casos estudiados. Se puede convertir el AFR en el MTBF haciendo:

$$MTBF = \frac{HorasAño}{AFR} \quad (3)$$

Adaptando esta idea al entorno Grid que nos ocupa y partiendo de las definiciones dadas en [68,69], podemos identificar y relacionar diferentes tipos de fiabilidad a distintos niveles y para distintos tipos de recursos:

**Fiabilidad de Programa Grid.** Probabilidad de que un programa dado se ejecute satisfactoriamente sobre múltiples nodos virtuales, incluyendo el intercambio de información a través de enlaces virtuales con recursos remotos, en el ámbito de un sistema de computación Grid.

**Fiabilidad de Sistema Grid.** Probabilidad de que todos los programas involucrados en un sistema Grid concreto se ejecuten correctamente.

**Fiabilidad de Servicio Grid.** Probabilidad de que todos los programas relacionados con un determinado servicio se ejecuten correctamente para dar el servicio requerido. Un servicio Grid puede implicar la ejecución de varios programas que utilicen recursos distribuidos en la Grid. Esta fiabilidad es similar a la de un sistema Grid, pero considerando únicamente los programas implicados en el servicio.

**Fiabilidad Grid.** Probabilidad obtenida al incorporar varios aspectos de la infraestructura Grid, incluyendo el sistema de gestión de recursos, la red de comunicación, así como los recursos hardware y software integrados en la misma. En primera instancia está determinada principalmente por la fiabilidad del sistema de gestión de recursos, aunque también hay una componente muy importante relacionada con la fiabilidad de la red de comunicación. Si se quiere mayor nivel de detalle, habrá que integrar también la fiabilidad de los recursos software y hardware involucrados.

Nuestro modelo propone reutilizar las fórmulas que se proponen en [68,69] para calcular todos estos tipos de fiabilidad, pero también estamos diseñando un sistema de medida independiente de los métodos utilizados para calcular los valores concretos de fiabilidad para cada tipo de recurso Grid. Dado que existen múltiples variables que se utilizan en el cálculo de cada tipo de fiabilidad y que puede suceder que los proveedores de información no ofrezcan algunas de ellas o que no utilicen las mismas fórmulas para obtenerlas, estamos diseñando una red Bayesiana que nos permita aprender cómo se relacionan las variables usadas en las definiciones anteriores.

Actualmente estamos definiendo qué variables vamos a utilizar, cuál es el rango de valores o fórmula (distribución) que representa a cada una de ellas, y las relaciones que vamos a establecer entre ellas a partir de las definiciones dadas en [68,69]. Al ejecutar la red Bayesiana con valores experimentales obtendremos qué variables están más relacionadas entre sí y cuáles pueden ser estimadas a partir de otras. Si no disponemos de proveedores de información que nos aporten valores de fiabilidad de un recurso, éste se podrá estimar a través de la red.

Estos son los acrónimos y relaciones entre conceptos de fiabilidad que usaremos para diseñar la red Bayesiana: Resource Management Service failure ratio ( $\lambda_{RMS}$ ), Service time ( $t_s$ ), Size Information Exchanged / Load network connection (D), Network bandwidth / Speed (S), Communication time ( $T_c$ ), Link failure ratio ( $\lambda_{i,j}$ ), Host client failure ratio ( $\lambda_i$ ), Host server failure ratio ( $\lambda_j$ ), Run time program ( $t_m$ ), Usage time program ( $t_u$ ), RMS reliability ( $R_{RMS}$ ), Link reliability ( $R_L$ ), Communication reliability ( $R_c$ ), Host program reliability ( $R_{prog}$ ), Host resource reliability ( $R_{res}$ ), GridResource program reliability ( $R_{GPR}$ ), GridNode system reliability ( $R_S$ ), GridResource program reliability with RMS reliability ( $R_{GPR-RMS}$ ), GridNode system reliability with RMS reliability ( $R_{S-RMS}$ ).

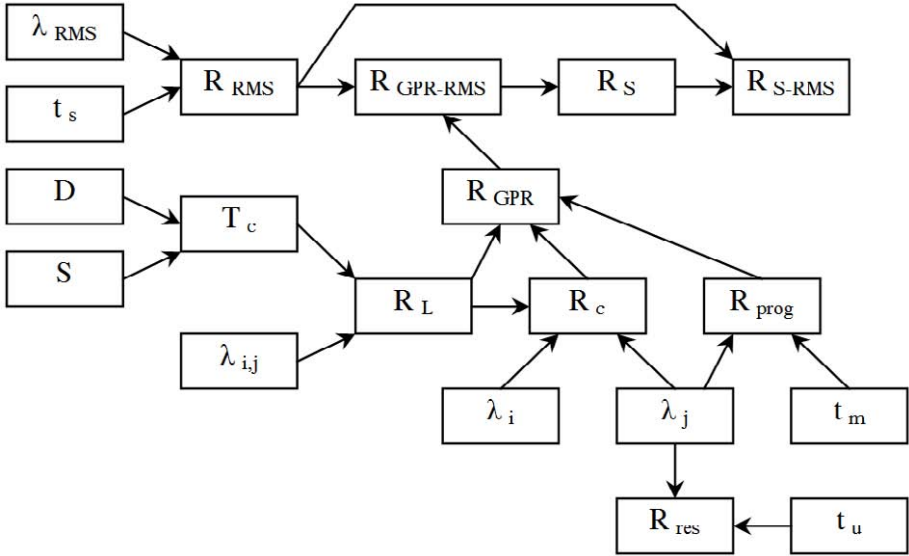


Fig. 5. Relaciones entre valores que permiten calcular distintos tipos de fiabilidad.

### 4 Conclusiones y Trabajo Futuro

Hasta la fecha, la mayoría de las iniciativas que hemos estudiado en el estado del arte carecen de extensibilidad o completitud. En la mayoría de los casos ni siquiera plantean la medición de las propiedades no funcionales, o ésta es muy trivial y no deja de ser un factor de corrección un tanto aleatorio y con poco peso. En general, no están esquematizadas, no se ha llevado a cabo un análisis detallado, o no ofrecen un modelo común para los recursos y sus propiedades.

Por lo tanto, el presente trabajo plantea un nuevo enfoque para modelar propiedades no funcionales sobre recursos Grid, así como un sistema de medida de dichas propiedades que luego permita a las potenciales herramientas o aplicaciones Grid proveedoras y/o consumidoras de este tipo de información utilizar un mismo lenguaje y un mismo esquema de referencia. Además, también se pretende demostrar que se puede alcanzar un descubrimiento, y posterior selección de recursos, más eficaz y efectivo basado en la descripción semántica de los recursos Grid.

En este artículo hemos presentado brevemente un modelo híbrido basado en una ontología para representar recursos Grid y propiedades no funcionales o de calidad de servicio asociados a los mismos, y en un sistema de medida de dichas propiedades. Se trata de un modelo que se está desarrollando para promover el consenso en la representación y medición/cálculo de propiedades no funcionales de recursos en la Grid y que pretende ser suficientemente genérico para que pueda ser reutilizado en todo tipo de Grids.

Actualmente se está desarrollando el modelo prestando especial atención a la clasificación de los tipos de recursos que pueden existir en la Grid y a las propiedades no funcionales más importantes y que más referencias tienen en la literatura del área: disponibilidad (availability), rendimiento (performance) y fiabilidad (reliability), siendo conscientes de que existen muchos conceptos de QoS que aún no hemos considerado. El modelo no solo propone cómo representar esta información, sino también cómo obtenerla y medirla.

El desarrollo del modelo de representación está basado en ontologías, una tecnología que actualmente se utiliza para modelar gran cantidad de dominios de aplicación y para la que existen una gran cantidad de herramientas. Hemos escogido desarrollar nuestra ontología en OWL porque es uno de los lenguajes más extendidos e incluso está considerado el lenguaje estándar de facto de la Web Semántica.

El middleware de la Grid es abundante y existen muchas herramientas que se pueden utilizar como proveedores de información sobre los recursos que se encuentran desplegados en un entorno Grid y otras tantas aplicaciones potenciales consumidoras de esta información que podrían ver mejorada su funcionalidad gracias a la innovación que se pretende desarrollar con el presente trabajo.

Aunque aquí no se ha presentado, hemos realizado un estudio de los proveedores de información Grid existentes, así como de los posibles consumidores de información Grid. Como resultado, hemos comprobado que algunas de las herramientas Grid mencionadas anteriormente en el estado del arte pueden ser utilizadas para desarrollar nuestro sistema de medida extendiéndolas con nuestros propios scripts para agregar o procesar datos que permitan obtener y representar en nuestro modelo los valores de QoS que nos interesan.

Además, como resultado del estudio de los posibles consumidores de información, hemos decidido probar nuestro trabajo con el metaplanificador GridWay, ya que se ajusta perfectamente a nuestros objetivos. Hemos estudiado sus características y actualmente estamos diseñando un nuevo algoritmo de selección de recursos basado en criterios de propiedades QoS de los mismos que pueda ser incluido entre los algoritmos de selección de recursos de GridWay.

## Referencias

1. David De Roure and York Sure. Semantic Grid - The convergence of technologies. *Journal of Web Semantics*, 4(2):82-83, 2006.
2. John Brooke, Donal Fellows, Kevin Garwood, and Carole Goble. Grid Computing, volume 3165/2004 of *Lecture Notes in Computer Science*, chapter Semantic Matching of Grid Resource Descriptions, pages 240-249. Springer, 2004.
3. Oscar Corcho, Pinar Alper, Ioannis Kotsiopoulos, Paolo Missier, Sean Bechhofer, and Carole Goble. An overview of S-OGSA: A reference Semantic Grid architecture. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(2):102-115, June 2006.
4. Ganglia. Ganglia distributed monitoring system, <http://ganglia.sourceforge.net>, 2008.
5. Nagios. Nagios distributed monitoring system, <http://www.nagios.org>, 2008.
6. GridICE. Gridice distributed monitoring system, <http://gridice.forge.cnaf.infn.it>, 2008.
7. Hawkeye. Hawkeye monitoring system, <http://www.cs.wisc.edu/condor/hawkeye/>, 2008.
8. CluMon. Clumon distributed monitoring system, <http://clumon.ncsa.uiuc.edu>, 2008.
9. INCA. Inca distributed monitoring system, <http://inca.sdsc.edu/drupal>, 2008.

10. MonaLISA. Monalisa distributed monitoring system, <http://monalisa.caltech.edu>, 2008.
11. Globus. Globus middleware, <http://www.globus.org/toolkit/about.html>, 2008.
12. L. Bitonti, T. Kiss, G. Terstyanszky, T. Delaitre, S. Winter, and P. Kacsuk. Dynamic testing of legacy code resources on the Grid. In Conference proceedings of the ACM International Conference on Computing Frontiers, number ISBN 1-59593-302-6, pages 261-268, Ischia, Italy, May 2-5 2006.
13. Askalon. SCALEA-G and Askalon web site, <http://www.askalon.org>, 2008.
14. GRASP. Grasp web site, <http://grail.sdsc.edu/projects/grasp>, 2008.
15. Gondor. Condor-g resource manager, <http://www.cs.wisc.edu/condor/condorg>, 2008.
16. PBS. PBS/Torque resource manager, <http://www.clusterresources.com/pages/products/torque-resource-manager.php>, 2008.
17. Sun. Sun Grid Engine resource manager, <http://www.sun.com/software/gridware>, 2008.
18. Platform. LSF resource manager, <http://www.platform.com/products/platform-lsf>, 2008.
19. Cluster Resources. Maui resource manager, <http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php>, 2008.
20. OASIS. WSRF, <http://www.globus.org/wsrp>, 2008.
21. GridWay. GridWay meta-scheduler web site, <http://www.gridway.org>, 2008.
22. Askalon. GridARM Askalon's grid resource management system, <http://www.dps.uibk.ac.at/projects/brokerage>, 2008.
23. g-LITE CERN. LCG/g-LITE broker web site, <http://glite.web.cern.ch/glite/packages/r3.0/deployment/lcg-rb/lcg-rb.asp>, 2008.
24. Rajkumar Buyya, David Abramson, and Jonathan Giddy. Nimrod-G: An architecture for a resource management and scheduling system in a global computational grid. pages 283-289. Press, 2000.
25. John Brooke, Donal Fellows, and Jon MacLaren. Resource brokering: The EuroGrid/GRIP approach. In All hands in UK, 2004.
26. SRB. Storage Resource Broker web site, <http://www.sdsc.edu/srb/index.php>, 2008.
27. NGS. NGS resource broker, <http://www.grid-support.ac.uk/content/view/205/196>, 2008.
28. Moab. Moab Grid scheduler web site, <http://wb.clusterresources.com/pages/products/moab-grid-suite.php>, 2008.
29. EGEE. EGEE Workload Manager Service (WMS) project web site, <http://glite.web.cern.ch/glite/packages/r3.0/deployment/glite-wms/glite-wms.asp>, 2008.
30. GridLab. GRMS (Grid(Lab) Resource Management) web site, <http://www.gridlab.org/workpackages/wp-9>, 2008.
31. Gridbus. Gridbus Grid Service broker web site, <http://www.gridbus.org/broker>, 2008.
32. NorduGrid. Advanced Resource Connector (NorduGrid's ARC) web site, <http://www.nordugrid.org/middleware>, 2008.
33. Globus. WebMDS web site, <http://www.globus.org/toolkit/docs/4.0/info/webmds>, 2008.
34. P-GRADE. P-GRADE Grid portal, <http://www.lpd.sztaki.hu/pgrade>, 2008.
35. Globus. CSF web site, <http://www.globus.org/toolkit/docs/4.0/contributions/csf>, 2008.
36. WebCom-G. WebCom-G web site, <http://www.webcom-g.org>, 2008.
37. GPT. Grid Packaging Tools, <http://www.gridpackagingtools.org>, 2008.
38. HPC4U. HPC4U web site, <http://www.hpc4u.org>, 2008.
39. OGF. GLUE schema working group (GLUE-WG), <http://forge.ogf.org/sf/projects/glue-wg>, 2008.
40. OGF. EGA reference model (EGA-RM), <http://forge.ogf.org/sf/projects/rm-wg>, 2008.
41. OGF. OGSA working group (OGSA-WG), <http://forge.ogf.org/sf/projects/ogsa-wg>, 2008.
42. OGF. Reference Model working group (RM-WG), <http://forge.ogf.org/sf/projects/rm-wg>, 2008.
43. OGF. OGSA Basic Execution Services working group (OGSA-BES-WG), <http://forge.ogf.org/sf/projects/ogsa-bes-wg>, 2008.
44. OGF. Job Submission Description Language working group (JSDL-WG),

- <http://forge.ogf.org/sf/projects/jsdl-wg>, 2008.
45. OGF. Information Modelling in OGSA, <http://forge.ogf.org/sf/projects/rm-wg>, 2008.
  46. OGF. CIM based Grid schema working group (CGS-WG), <http://forge.ogf.org/sf/projects/cgs-wg>, 2008.
  47. K-Wf-Grid. GOM Grid ontologies, <http://gom.kwfgrid.net/web/space/grid+ontologies>, 2007.
  48. M. Parkin, S. van der Berghe, O. Corcho, D. Snelling, and J. Brooke. Knowledge of the Grid: a Grid resource ontology. In Marian Bubak, Michal Turala, and Kazimierz Wiatr, editors, CGW'06 Proceedings, 2006.
  49. OGF. OGF work groups web site, <http://forge.ogf.org/sf/projects/>, 2007.
  50. GSMO. GSMO web site, <http://www.gsmo.org>, 2008.
  51. Robert M Pirsig. MOQ web site: [www.moq.org](http://www.moq.org), 2007.
  52. Alfonso Sánchez-Macián, Luis Bellido, and Encarna Pastor. Ontologías para la medida de la calidad de servicio percibida. In Libro de Ponencias de las V Jornadas de Ingeniería Telemática (JITEL 2005), pages 693-700, 2005.
  53. Vladimir Tomic, Kruti Patel, and Bernard Pagurek. WSOL - A language for the formal specification of classes of service for web services. In CSREA Press, editor, Proceedings of the 2003 International Conference on Web Services (ICWS03), pages 375-381. CSREA Press, 2003.
  54. FIPA. FIPA-QoS web site, <http://www.fipa.org/specs/fipa00094>, 2002.
  55. Shonali Krishnaswamy. WS-QoS web site: [www.csse.monash.edu.au/~shonali/ws-qos/](http://www.csse.monash.edu.au/~shonali/ws-qos/), 2003.
  56. Zhou Chen. OWL-QoS, <http://www.ntu.edu.sg/home5/pg04878518/owlqosontology.html>, 2005.
  57. E.M. Maximilien and M.P Singh. Toward autonomic web services trust and selection. In ACM Press, editor, Proceedings of 2nd International Conference on Service Oriented Computing (ICSOC), pages 212-221, New York, Nov 2004.
  58. Glen Dobson, Russell Lock, and Ian Sommerville. QoSOnt: A QoS ontology for service-centric systems. In EUROMICRO '05: Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications, pages 80-87, Washington, DC, USA, 2005. IEEE Computer Society.
  59. W3C. OWL features, <http://www.w3.org/tr/owl-features>, 2008.
  60. DIRC. DIRC project web site, <http://www.dirc.org.uk>, 2008.
  61. SeCSE. SeCSE project web site, <http://secse.eng.it>, 2008.
  62. David Buján-Carballal, Oscar Corcho-García, and Josuka Díaz-Labrador. A model of non functional properties for Grid resources. In 3rd CoreGRID Workshop on Grid Middleware, Barcelona, June 5th - 6th 2008.
  63. David Buján-Carballal. NFP4Grid owl file. <http://paginaspersonales.deusto.es/dbujan/NFP4Grid.owl>, 2007.
  64. RAE. Fiabilidad en RAE, [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=fiabilidad](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=fiabilidad), 2008.
  65. Wikipedia. Fiabilidad en Wikipedia, <http://es.wikipedia.org/wiki/Fiabilidad>, 2008.
  66. IEEE. Reliability in IEEE, <http://www.ieee.org>, 2008.
  67. Imperial College of London. Free on-line dictionary of computing, 2007.
  68. Y.S. Dai, M. Xie, and K.L. Poh. Computing System Reliability, chapter Reliability of Grid Computing Systems, pages 179-205. Springer US, 2004.
  69. Y. S. Dai, M. Xie, and K. L. Poh. Reliability of Grid service systems. *Comput. Ind. Eng.*, 50(1):130-147, 2006.