A model of non functional properties for Grid resources

David Buján-Carballal[1], Oscar Corcho-García[2], Josuka Díaz-Labrador[1]

[1] Department of Software Engineering,
Faculty of Engineering,
University of Deusto,
Avenida de las universidades 24,
48007 Bilbao, Vizcaya (Spain)
dbujan@eside.deusto.es, josuka@eside.deusto.es

[2] Department of Artificial Intelligence,
Faculty of Informatics
Polytechnic University of Madrid,
Campus de Montegancedo s/n,
28660, Boadilla del Monte, Madrid (Spain)
ocorcho@fi.upm.es

Abstract. So far, Grid information providers basically give functional values about resources, although some of them also provide aggregated information. Therefore, existing Grid information models essentially represent this syntactic information and also propose different taxonomies of resources. Hence, Grid information consumers commonly use functional properties to select resources in order to send jobs to a Grid. There are some approaches that try to use isolated techniques to take into consideration some QoS properties, like performance. In this paper, we propose a unified model for representing Grid resources and their non functional properties, adding semantics to Grid information systems. On one hand, this model is an ontology-based model developed to integrate existing approaches of Grid information models and non functional properties representations in general. But on the other hand, our model also proposes a measuring system - currently in development - for some non functional properties like reliability, availability and performance. Here we only present an overview about how to represent and measure reliability information of resources in Grids. This example is used to illustrate our work in progress.

Keywords: Semantic Grid, Grid Information Systems, Non functional properties, Ontology-based models, Measuring systems.

1 Introduction

One of the open issues in the Grid is the discovery of services and resources when Grid users or applications send jobs to the Grid in order to be executed. Most of the approaches are based on the analysis of functional properties about resources, that is, those characteristics that define what is what the resource makes or offers. However, there are non functional properties (like reliability, availability, performance and so on) that can be used in this context. Non functional properties specify global restrictions that must be satisfied [1], indicating how the resource operates or how it exposes its functionality [2].

Sometimes the jobs sent to the Grid fail because the assigned resources to them were not available or had fallen. In these cases, sequential or parallel jobs must be sent back again to the Grid. It implies a new selection and allocation of resources, as well as a waste of time, work and resources, that increases the global time to carry out a job.

On one hand, there are Grid tools for resuming some kind of jobs from the point of failure, but this is not a general case and besides it does not solve the problem of wasting time and work. And on the other hand, fault-tolerant systems based on recovery techniques are very costly in distributed systems, so those jobs that have not finished after a time out are cancelled and sent back again to the Grid. Therefore, one of the most used solutions consists in monitoring the resources that fail and putting them manually as requirements for the job sent to the Grid. Thus, the job is never sent to those "bad" resources. It is usually also specified an expected time for carrying out the job. This requirement let a job to be assigned a resource of which response time is smaller than expected time to finish for the given job. Besides, it is usually applied the same idea to other concepts like the size of files used by a job or its expected output, regarding the available physical memory space into the storage elements resources.

However, previous solutions are ad-hoc and we think that non functional properties could be useful in these cases. This is an issue in the field of the Grid that has not been studied in depth yet.

Many Grid middleware services (aka Grid information consumers) require knowledge about the behavior of Grid resources in order to effectively select, compose, and execute jobs in dynamic and complex Grid systems. To provide QoS or non functional information about resources for building such knowledge, Grid tools (aka Grid information providers) have to select, measure, and analyze various QoS metrics of resources (about their behavior executing jobs, storing data, etc.). However, there is a lack of a comprehensive study of QoS metrics which can be used to evaluate the QoS of a resource (its non functional properties).

Moreover, given the complexity of both Grid systems and QoS concepts, semantics of existing Grid resources and their essential QoS-related concepts and relationships should be well described. There are several efforts in order to classify resources in the Grid and QoS concepts in SOA, but there is no link among them.

On the one hand, existing Grid information models don't use the same language to represent the same concepts about resources, although some research groups have detected a need for integrating and making their models interoperable. Besides, there is a lack of representation of non functional properties about resources. At best, some models try to represent performance measures about resources, but there is a lack of details about non functional properties in general. Some Grid information consumers and providers use isolated techniques to deal with this information.

On the other hand, most existing work in QoS models concentrates on business workflows and Web services processes in SOA, but several metrics targeted in these models are not valid for Grid environments because Grid resources and their non functional properties are more diverse, dynamic, and inter-organizational. That's why a new model with new metrics is needed.

However, we can learn good lessons from these two groups of models in order to aggregate isolated techniques used by information providers and consumers in both fields. Some of the goals of our future work are to integrate existing approaches into Grid information systems and to enrich the discovery and selection of resources in the Grid using semantic information associated to them [3]. We want to contribute to the Semantic Grid idea: the Semantic Grid is an extension of the current Grid in which information and services are given well-defined meaning through machine-processable descriptions which maximize the potential for sharing and reuse [4,5].-

2 Background

2.1 Grid information consumers

So far, most of these Grid information consumers commonly use functional properties to select resources in order to send jobs to a Grid. For example, many Grid tools or applications that carry out functions of planning, selection and/or resource management in the Grid (schedulers, meta-schedulers, resource brokers and so on) discover resources using properties like characteristic of the processor, size of the memory, space of available storage, software that is found installed, response time to the requests of service and so on.

Some of the following initiatives could be good candidates for using non functional properties in their algorithms of selection: Meta-scheduler GridWay, GridARM, LCG/g-LITE, Nimrod-G, EUROGRID/GRIP, SRB, NGS Resource Broker, Moab Grid Scheduler (aka Silver), EGEE WMS, GRMS, Gridbus Grid Service Broker, NorduGrid's ARC. Other Grid middleware that could consume non functional properties would be: WebMDS, Grid User Interfaces (Grid Portals like P-GRADE), Grid tools based on CSF, N1 Grid Engine, WebCom-G, GPT, HPC4U Middleware, or any WSRF client looking for resources (like GEMLCA or the NWS).

There are some approaches that try to use isolated techniques to take into consideration some QoS properties, like performance, but in a very ad-hoc way. Other approaches are still investigating how to incorporate non functional properties in their algorithms of selection.

2.2 Grid information providers

Nowadays, Grid middleware is abundant and there are many monitoring tools that can be used as providers of information about Grid resources: Globus middleware (WS GRAM, MDS4, RFT, RLS, CAS), monitoring systems (Ganglia, Hawkeye, Inca, MonALISA, GEMLCA-GMT, GridICE, GRASP, CluMon, SCALEA-G and other Askalon monitoring tools for their Resource Broker and Scheduler), queueing systems

(Condor-G, Sun Grid Engine, Torque/PBS, LSF, Maui) or any other WSRF service that publishes resource properties.

2.3 Grid information models

Regarding the representation and use of quality of service in general, there are some references that describe general and philosophical aspects to be considered, but don't provive a model: MOQ [6]; QoS with QoE and QoBiz [7]; Tosic et al [8]. But there are also other works that have implemented ontologies to represent QoS information: FIPA-QoS [9]; WS-QoS [10]; Zhou Chen's OWL-QoS (previously DAML-QoS) [11]; Maximilien and Singh's OWL-QoS [12]; QoSOnt [13] is trying to unify all the previous initiatives.

As far as Grid is concerning, there are some OGF research groups [14] dealing with aspects related to the representation of information about the Grid: GLUE-WG, EGA-RM, OGSA-WG, RM-WG, OGSA Basic Execution Services WG (OGSA-BES-WG), JSDL-WG, Information Modelling in OGSA, CGS-WG, GOM [15,16], Grid Ontology [17]. There are other OGF's groups of which work could be useful in this area, but their activity has no repercussions yet or has just began: OGSA-RSS-WG, SRM, GRIDREL-RG, TC-RG, NM-WG, GB-RG, RUS-WG, UR-WG, GCE-RG, SEM-RG.

Some of these research groups suggested in the past Open Grid Forums the need of unifying criteria and developments to make these different models interoperable or, at least, to converge in a same more generic specification, since every group tackle the modelling from its own cases of use and goals. Next, we describe most significant information models briefly.

2.3.1 An analysis framework for resources and non functional properties models

So far, previous approaches have some limitations that make it difficult to adopt them directly into our model. Despite providing mechanisms for QoS specification, these models lack the flexibility provided by an ontology for dealing with the semantic meaning of QoS constraints. In many cases they are not extensible or adaptable to the Grid. In other cases, they aren't comprehensive solutions, focusing on functional properties only or providing an ad-hoc solution for some isolated QoS properties. In these cases, some of them don't suggest how to measure the non functional properties or they use very trivial and random criteria. In general, there is no schematic or detailed analysis of Grid resources and their non functional properties. Besides, there is no common Grid information model yet.

Next, we summarize the analysis done using this comparison framework: Comprehensiveness (detail level representing types of Grid resources and non functional properties), Measurement (methods used for measuring non functional properties, if represented), Standards / Technologies (how the model is implemented), Supporting application (information consumers or providers ready to use the model).

Table 1. Comparison framework for existing information models.

Information Models	Comprehensiveness	Measurement	Standards / Technologies	Supporting application
GLUE 1.1	Good taxonomy of resources, but not enough. There is no reference to non functional properties, except for some isolated QoS values.	None. Only define basic types (int, float, string) for properties.	UML diagrams, XML schema, Xpath 2.0	Integrated into GT 4.0.3. Ganglia and Hawkeye provide an output using this XML schema.
GLUE 1.3	Extended taxonomy of resources. There are new QoS values, but nothing else.	None. Only define basic types for new properties.	UML diagrams, XML schema, Xpath 2.0	It can be integrated into GT 4.0.x and Grid monitoring tools.
GLUE 2.0	Some new properties that could be used to represent some non functional values, but there is still no reference to a formal schema for them.	None. Only define basic types for new properties.	UML diagrams, XML schema, Xpath 2.0	It can be integrated into GT 4.0.x and Grid monitoring tools.
OGSA-BES	No taxonomy of resources. Some properties that could be used to represent some non functional values, but there is no reference to a formal schema.	None.	UML diagrams, Port-types. GLUE schema is generally considered as a specialisation of BES.	None.
JSDL	No taxonomy of resources. It describes resource requirements of the Grid jobs, but there is no significant reference to non functional requirements or properties.	None.	XML schema. Informed by DRMAA, CIM, and POSIX.	Core vocabulary informed by Condor, GT, LSF, PBS, SGE, and Unicore.
CIM	No taxonomy of resources. There is no formal description of non functional properties.	None.	UML diagrams.	None.
UR	No taxonomy of resources. It describes some properties about how the jobs use the resources that can be useful to calculate non functional properties.	Some ideas about how to measure them, but there is no proposal about how to do it in practice.	Usage properties and XML infoset representation.	None.

EGA	No taxonomy of resources. There is no formal description of non functional properties.	The model only gives slightly some ideas about how to measure in general some non functional properties.	Reference CIM or OGSA. There is an opportunity for the EGA to either unify models or at least make them consistent with each other.	None.
RM	There is no significant work done yet. It uses the EGA Reference Model and the OGSA Glossary as its starting points.	There is no significant work done yet.	This approach will try to use OWL, SML, XML Schemas.	None.
GOM	Useful taxonomy of resources and services. Performance property is defined deeply.	Focus on performance only, although we have no access to this information yet.	OWL.	Performance tools from Askalon framework and K-Wf Grid project.
GRO	Useful Grid foundational ontology. Bad and basic taxonomy of resources. Some references to QoS concepts.	None.	OWL.	None.
QoSOnt	No taxonomy of resources. No reference to the Grid. Useful taxonomy of QoS concepts and metrics.	None.	OWL.	None.

3 Our model

In Software Engineering, an ontology may be described as a specification of a conceptualization or a description of the concepts and relationships that may exist for an agent or a community of agents. Ontologies allow the sharing and reuse of knowledge. Software applications can be written according to the commitment of using the vocabulary defined by an ontology for a particular domain. That way, agents that share the same ontology can communicate among themselves since their vocabulary is understood by everyone.

In our approach, we propose a detailed classification of resources and services that can exist in the Grid, paying attention to multiple levels of abstraction. Taking into account models and specifications given by previous work groups, we propose an ontology for describing Grid resources and the most important non functional properties referenced in previous works: performance, availability and reliability. When different monitoring tools support a common ontology, the Grid users and applications can benefit from having a common understanding of Grid resources and their QoS behavior given by these tools. Our model – currently in development – also needs a formal measuring system for non functional properties. Each one has different characteristics (complexity of calculation, parameters of relevance, etc.).

We are defining, collecting and associating various metrics with relevant QoS concepts about resources in Grids. We are also designing a range for several metrics in order to make possible a comparison. Our study considers QoS metrics for software and hardware resources in many levels of detail such as systems, services, applications/programs, nodes, networks, hosts, etc., following a Grid resource model classification. So far, we only can present an overview about how to represent and measure reliability in Grids. We also introduce how the ontology can be utilized for representing and analyzing non functional properties of Grid resources.

3.1 Representing Grid resources and non functional properties

The ontology we are introducing in this paper is available from [20] for public inspection, use and comment. We have integrated some ideas from different ontologies and reused / extended some classes and created new ones to develop our own ontology. GOM Resource ontology (ResourceOntology.owl), GOM Service ontology (ServiceOntology.owl), Grid Foundational Ontology (Base.owl), Globus Ontology (Globus.owl) have been used as starting point for designing a more detailed and multilevel classification of Grid resources and services, taking into account the roles that can be associated to them in a Grid environment. Since these ontologies have a poor definition of non functional properties, we have used the QoSOnt ontology (QoSOnt2.owl) as starting point to develop our model. Due to space requirements, here we only present an overview of it with snapshots. You can navigate all the classes consulting NFP4Grid.owl file.

Image 1. Partial diagram of classes related to resources.

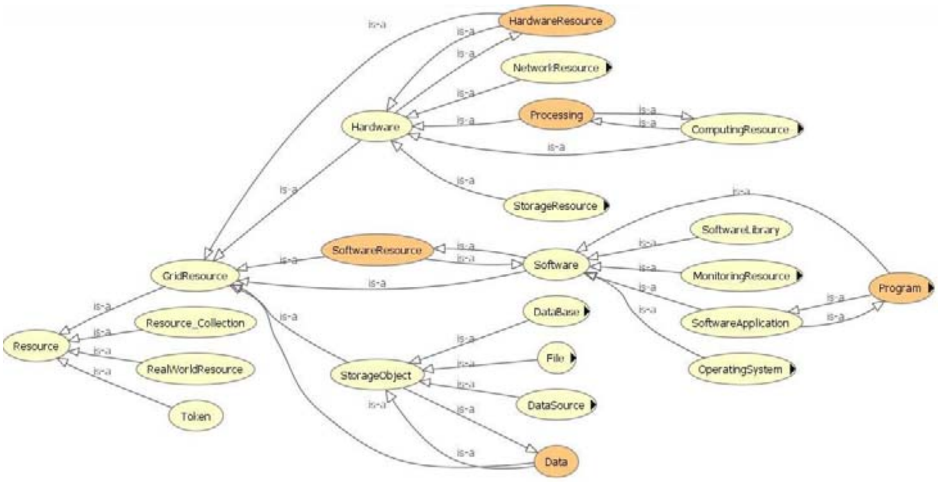


Image 2. Partial diagram of classes related to services.

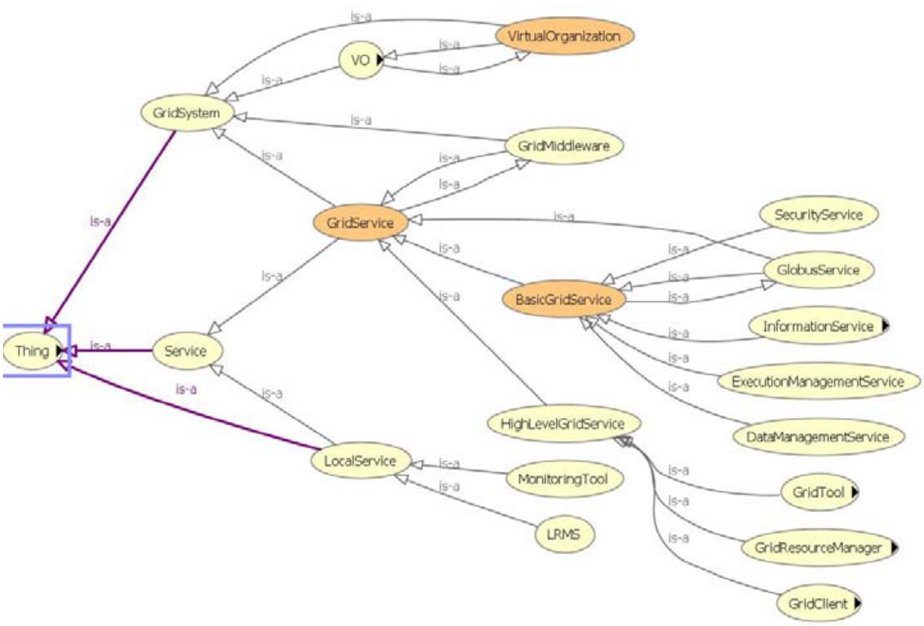
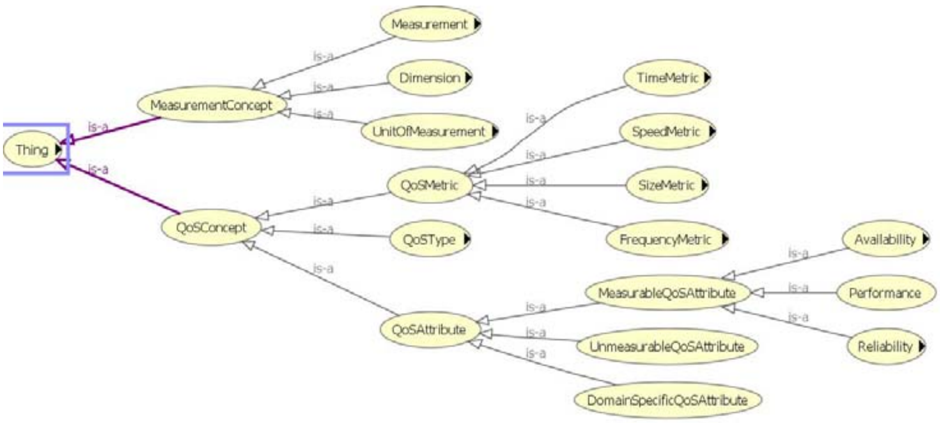


Image 3. Partial diagram of classes related to roles.

Image 4. Partial diagram of classes related to non functional properties.



Numerical information about non functional properties will be represented directly into the ontology, but also associated with some ranges of possible values, since languages as OWL cannot handle them directly. Thus, information consumers could extract values of these non functional properties of different represented resources in order to compare them. So our model has a hybrid approach, as it is based on an ontology and other techniques at the time of selecting the resources.

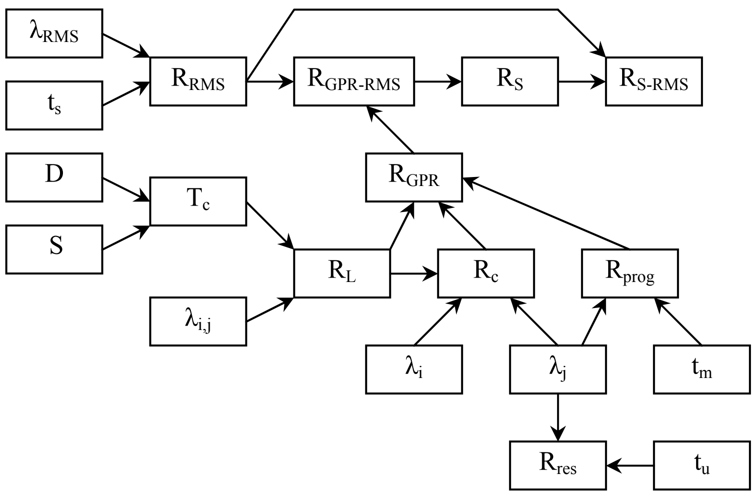
3.2 Measuring system for reliability

We can define reliability as "an attribute of any system that consistently produces the same results, preferably meeting or exceeding its specifications" [18]. However, we must define, relate and measure other types of reliability at different levels in a Grid environment. Our starting point are the definitions given by [19] for Distributed program reliability, Distributed system reliability, Distributed service reliability, Grid program reliability, Grid system reliability, Grid service reliability and Grid reliability. The last one can be defined as the probability obtained by incorporating various aspects of the grid structure including the resource management system, the network and the integrated software / resources.

We want to integrate all the components into the analysis, but we do not know exactly how they are related. We are still designing a measuring system based on a Bayesian network, since this technique can help us to learn how all the variables used in the previous definitions affect each other. Nowadays, we are setting the assumptions, variables and links of the Bayesian network, as well as the ranges or formulas (distributions) of the variables. Besides, this method let us to estimate some variables without having some others, even if they are a priori required inputs.

Next we introduce the acronyms and relations among reliability concepts we are using as starting point for designing the Bayesian network: Resource Management Service failure ratio (λ_{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 (λ_i), Host server failure ratio (λ_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}).

Image 5. Relations among reliability concepts.



4 Conclusions and future work

This work has briefly presented a model and reference ontology of non functional properties for Grid resources that not only capture concepts such as Grid resources and services but also non functional properties about them. We also introduce how they can be obtained from Grid information providers and used by Grid information consumers, using reliability as a proof of concept - currently in development. By taking this novel perspective of the non functional properties for Grid resources we have provided the foundations for describing Grid resources and their non functional properties.

Many QoS concepts are not considered in our approach yet. We observed a number of works building Grid information models and QoS metrics for Web services separately. Some of them have concepts and metrics in common with our model.

Some of the Grid tools mentioned before in section 2.2 can be used to develop our measuring system of not functional properties, although we will develop our own scripts as well. We will aggregate or process given data by these providers in order to calculate our target non functional properties and to represent them in our model. In fact, we are planing to design an aggregator resource source script. Its implementation will be based on MDS4 to collect, store and index information about resources, respond to queries concerning the stored information using the XPath language, and control the execution of testing and information retrieval tools built as part of the GT4 Basic Services Monitoring Toolkit. Afterwards, we are planning to develop a prototype system or plugin for a Grid information consumer (i.e. GridWay) in order to test our model. Our future work will concentrate on promoting the uptake and use of this model to provide advanced scheduling techniques based on the non functional properties of resources.

References

[1] Rosa, Nelson S. and Cunha, Paulo R. F. and Justo, George R. R. Process-NFL: A Language for Describing Non-Functional Properties. IEEE Computer Society Press (2002)

[2] Chung, Lawrence. Representing and Using Non-Functional Requirements: A Process-Oriented Approach. University of Toronto (1993)

[3] Semantic matching of Grid resource in Grid Computing. J.M. Brooke, D. Fellows, K. Garwood and C. Goble. Second European AcrossGrids Conference. Revised Papers Editors: Marios D. Dikaiakos, LNCS 3165, pp. 240-249. Jan. 2004.

[4] "Semantic Grid: The Convergence of Technologies." Dagsthul Seminar No 05271, organised by C. Goble, C. Kesselman, Y.Sure (2005)

[5] O. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, C. Goble. An overview of S-OGSA: a Reference Semantic Grid Architecture. Journal of Web Semantics 4(2):102-115. June 2006

[6] MOQ web site, http://www.moq.org

[7] Alfonso Sánchez-Macián, Luis Bellido, Encarna Pastor. Ontologías para la Medida de la Calidad de Servicio Percibida. Libro de Ponencias de las V Jornadas de Ingeniería Telemática (Jitel 2005), Vigo, Pontevedra, España, Septiembre 2005, pp. 693-700. ISBN 84-8408-346-2.

[8] Tosic, V., Pagurek, B., Patel, K. "WSOL — A Language for the Formal Specification of Classes of Service for Web Services". ICWS03. Las Vegas, USA. June 23–26, 2003, CSREA Press, pp. 375–381.
[9] FIPA-QoS specification, http://www.fipa.org/specs/fipa00094
[10] WS-QoS web site, http://www.csse.monash.edu.au/~shonali/ws-qos/index.html
[11] OWL-QoS web site, http://www.ntu.edu.sg/home5/pg04878518/OWLQoSOntology.html
[12] Maximilien, E.M., Singh, M.P. "Toward Autonomic Web Services Trust and Selection". Proceedings of 2nd International Conference on Service Oriented Computing (ICSOC), pages 212–221, New York, Nov. 2004. ACM Press. http://maximilien.org/publications/papers/2004/Maximilien+Singh04b.pdf
[13] Dobson, G.; Lock, R.; Sommerville, I. "QoSOnt: a QoS ontology for service-centric systems". Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference on, Vol., Iss., 30 Aug.-3 Sept. 2005. Pages: 80- 87. http://digs.sourceforge.net/publications.htm
[14] OGF Work Groups web page, http://forge.ogf.org/sf/projects/
[15] GOM Grid ontologies (K-Wf Grid project), http://gom.kwfgrid.net/web/space/Grid+Ontologies
[16] Hong-Linh Truong, Thomas Fahringer, Francesco Nerieri, Schahram Dustdar. "Performance Metrics and Ontology for Describing Performance Data of Grid Workflows, IEEE International Symposium on Cluster Computing and Grid 2005 (CCGrid2005), 1st International Workshop on Grid Performability, IEEE Computer Society Press, Cardiff, UK, 9 - 12 May 2005.
[17] Knowledge of the Grid: a Grid resource ontology. Parkin M, van der Berghe S, Corcho O, Snelling D, Brooke J. Cracow Grid Workshop 2006. Cracow, Poland, October 2006.
[18] Free On-line Dictionary of Computing, http://foldoc.org/index.cgi?query=reliability&action=Search
[19] Min Xie, Kim-Leng Poh, Yuan-Shun Dai. Reliability of Grid Computing Systems. Computing System Reliability. Pages 179-205. Springer US, 2004. ISBN: 978-0-306-48636-4.
[20] NFP4Grid owl file. http://paginaspersonales.deusto.es/dbujan/NFP4Grid.owl