

# Integrating the Common Information Model with MDS4

I. Díaz, G. Fernández, M.J. Martín, P. González, J. Touriño  
Computer Architecture Group  
Department of Electronics and Systems, University of A Coruña  
Campus de Elviña s/n, 15071 A Coruña, Spain  
{idiaz,mariam,pglez,juan}@udc.es

## Abstract

*The management and monitoring of static and dynamic resources is a key issue in Grid environments. Information models are an abstract representation of software and hardware aspects of these resources, a common and structured representation that allows intra- and inter-Grid interoperability. Among them, the Common Information Model (CIM), standardized by the DMTF, is an object-oriented and extensible model. The aim of this work is to provide a CIM-based query service for resource management information in Grid systems. This is achieved by publishing the CIM information on the Monitoring and Discovery System version 4 (MDS4) component of the Globus Toolkit, and implementing a query engine that supports queries with arbitrary navigation depth on a CIM instance tree.*

## 1. Introduction

A Grid system coordinates resources that are not subject to centralized control, using standard, open, general purpose protocols and interfaces, to deliver nontrivial qualities of service. There are several middleware tools that provide the needed infrastructure for Grid systems. The Globus Toolkit [8], based on the Open Grid Services Architecture (OGSA) [9], is the one used in this work due to its widespread adoption and use.

A key aspect in Grid environments is the management and monitoring of the resources. Their heterogeneity and the need of interoperability between different middleware solutions for Grids, explain the need of a common model of information in order to allow the exchange of resource information both inside a Grid and across Grids.

An information model is an abstraction of real-world entities into formal constructs that can be represented in computer systems. It is not tied to any particular implementation, and it is used to exchange information between different domains.

Among the existing models of information for Grids' management, particular attention is paid in Grid literature to GLUE (Grid Laboratory Uniform Environment) Schema [1] and to CIM (Common Information Model) [6]. Both of them are object-oriented information models. The main purpose of the GLUE Schema is to define a common resource information model to be used as a base for the Grid Information Service (GIS), both for resource discovery and for monitoring activities. CIM is a more complex model, a unified model covering an extensive area of general and distributed computing. In this paper we present an infrastructure integrating CIM and MDS, that allows to access and query CIM information in a Grid system, enabling the definition of searches with established preconditions across an arbitrary number of object associations. The developed infrastructure can be used for resource discovery, fault diagnosis, scheduling, accounting and many other applications.

The structure of this work is as follows. Section 2 details the integration of CIM with the MDS information service in Globus. Section 3 presents a Grid service supporting queries on the CIM data. Section 4 covers related work. Finally, Section 5 concludes the paper.

## 2. Aspects of the Integration of MDS4 and CIM

This section covers the integration of the CIM model into the MDS4 component of Globus. Figure 1 shows an overview of this integration. In the left side the Globus middleware stores CIM data encoded into XML and communicates with a query service in the right, which retrieves CIM instances from MDS4 and, based on the CIM schema and its internal representation, conducts searches on these instances.

The Globus Monitoring and Discovery System version 4 (MDS4) is the component of the Globus Toolkit tasked with publishing the information needed for resource discovery and monitoring functionality. Unlike previous versions, MDS4 is based on WSRF services instead of on an LDAP

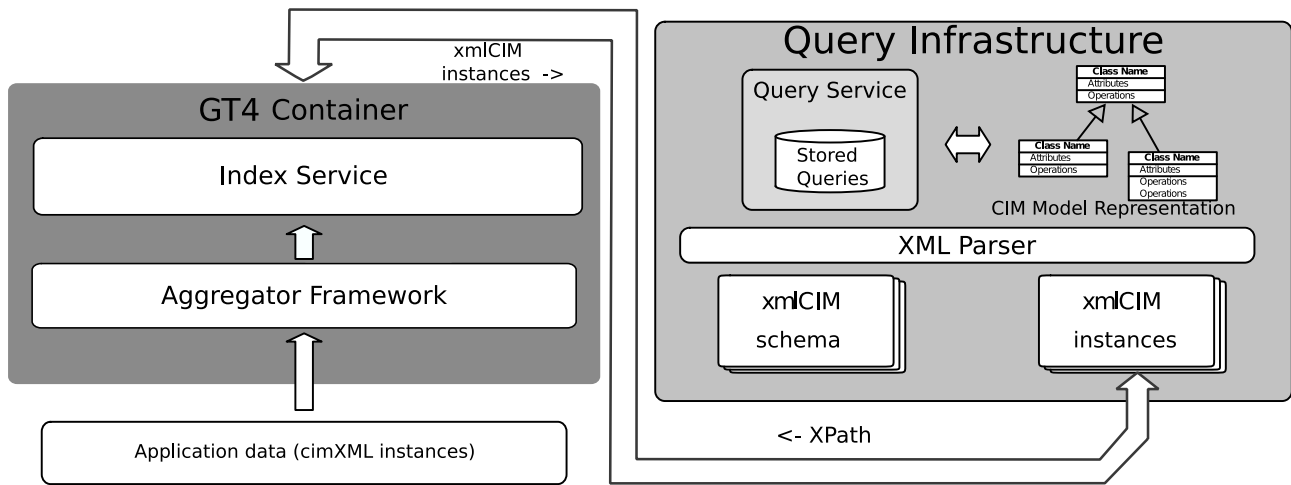


Figure 1. Overview of the CIM-MDS4 integration and query services.

directory. MDS4 is sometimes defined as having a “hour-glass” structure, as shown in Figure 2, in which the sources of information at the base communicate with “sinks” (or information users) at the top via a “narrower” middle section which represents both the MDS web service interface and the common information schemas used in the stored information. Thus, MDS4 does not have a standard schema (the information must be encoded in XML, though), but sources and sinks usually need to agree on one schema to communicate. Our approach uses CIM as a standard schema for Grid management applications.

CIM [6] is a management standard proposed by the Distributed Management Task Force (DMTF) to define an object-oriented and user-extensible information model for all management data: physical attributes like location, cabling, component placing, temperature and voltage; and logical abstractions such as users, software packages and dependencies, administration policy and hardware capabilities. The completeness and standardization of CIM as information model have caused its widespread use as a base for management ontologies. CIM can be represented in various formats, including xmlCIM [5], the representation in XML format used by the DMTF.

CIM information is contained in classes organized in a hierarchy based on inheritance. Each instance of a CIM class is identified by a unique name, formed from an aggregation of the value of some of its properties, called *key* properties. The CIM model also defines an association model which relates instances. Associations are first-class CIM entities which contain regular properties and referential properties. These referential properties point to regular instances by having the unique name of these instances as their value.

Associations can have 1-to-1, 1-to-many and many-to-

many cardinality. The 1-to-many subtype includes *weak* associations, which represent asymmetric associations where one end has no separate existence/name without the other.

The association model is central to CIM: it renders CIM object trees into navigable networks of management data, representing many kinds of useful information. For instance, it can relate an internet address in a virtual LAN with a card in a connector switch in a physical rack and give us its temperature, location and service date.

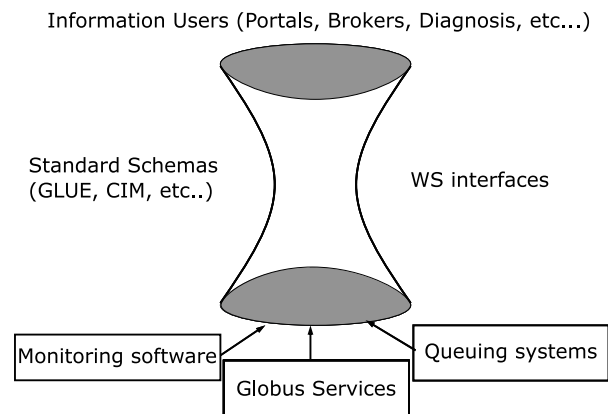


Figure 2. MDS4 “hourglass” representation.

MDS4 comprises two main services: the *Index Service* publishes, updates and manages the life cycle of resource information, and the *Aggregator Framework* defines three mechanisms for source applications to publish their data:

- The `QueryAggregatorSource` polls periodically the values of several WSRF properties.

- The `SubscriptionAggregatorSource` registers itself as the listener of a WS-Notification subscription, normally linked to a WSRF property.
- The `ExecutionAggregatorSource` invokes code which periodically returns XML-encoded information.

Encoding the entire CIM model as WSRF properties would be cumbersome and inflexible, because it would require to define all classes separately and to create code for their implementation, and to regenerate this code if new classes were added. An XML representation avoids these problems, and also it does not cause breakage if clients are not recompiled each time the model is changed. For this reason, the chosen approach was to use the `ExecutionAggregatorSource` to invoke `xmlCIM`-generating code.

The CIM code is encoded in `xmlCIM` with some minor alterations. The most significant one is that the schema name is encoded as part of the CIM class name, in order to have a flat namespace without name collisions. Also, method declaration, qualifiers and other information not relevant to instance description (i.e. not belonging to properties and naming information) are not included.

### 3. Query Support for Grid Management Applications

The publication of the CIM information as described in the previous section can be consumed directly by grid management applications, but without a query service to retrieve information selectively, the retrieval would not scale well for large grids.

The query service presented here can retrieve instances of a class filtered by a set of conditions on their properties, and/or can be reached following an arbitrary number of associations from all the instances in a set (which can be also predicated). This can be used to specify very expressive queries, and it is very convenient for users, which can easily formulate queries by establishing conditions on associated nodes. This section will first detail the internal representation of the CIM model used in the service in order to explain the query representation and execution.

#### 3.1. Model internal representation

Figure 3 shows the UML model of the CIM internal representation in the query service. As can be seen in the figure, the CIM information is modelled using a metaschema approach, so CIM classes are treated as instances of the `CIMClassInformation`, as opposed to

being translated directly as classes in the internal representation. This has the advantage of being much more flexible when new CIM classes are introduced (as only new instances of `CIMClassInformation` have to be considered), and easy to implement. The drawback is the additional overhead, since some implicit object qualities like the number and type of its properties have to be represented as objects and validation has to be explicit. CIM defines various primitives, such as schemas, classes, properties, indications, triggers, associations, qualifiers, methods and references. Schemas are encoded as part of the class name, as was described in Section 2. Indications, triggers and methods are excluded, since they are not needed for resource information querying.

Qualifiers are not treated equally; the most important ones — like those indicating the name of the superclass, whether a property is a key, if a class is abstract, and so on — are mapped as properties of `CIMClassInformation` or `CIMProperty`. Non-critical qualifiers are represented with the `CIMQualifier` class. This mapping avoids having to access a separate instance each time an important qualifier is used, but still allows to define new qualifiers.

References are represented by a special class named `CIMReference` instead of being represented by properties. Inheritance is explicit, with each class inheriting properties and key properties by direct copy from its parent.

The types of the properties are simplified to standard Java types: integer values are converted to Java `long`, real number to `double`, strings and characters to the `String` class, and finally, dates and times to `java.util.Calendar`.

Summarizing, the internal representation of the model has been devised trying to improve both performance and flexibility, and also to conserve expressive power.

#### 3.2. Query representation and execution

The query mechanism shown here supports a search for instances of a class filtered by any number of value restrictions on their properties. These classes can be further restricted by being reachable from certain instances navigating CIM associations. This formulation of the queries aims to reduce the complexity for the end user and to make the formulation of complex queries as intuitive as possible.

Figure 4 shows the UML representation of the queries, as a tree where the leaves are `SearchPathCustomTO`, a specification of a CIM instance and some properties whose values must be compared. The non-leaf nodes are `OperatorCustomTO`, a boolean operator that takes two nodes as parameters. An example of this configuration can be seen in Figure 5. The expression corresponding to this tree searches computers with a processor speed of 2Ghz or more, Linux operating system installed and which are cur-

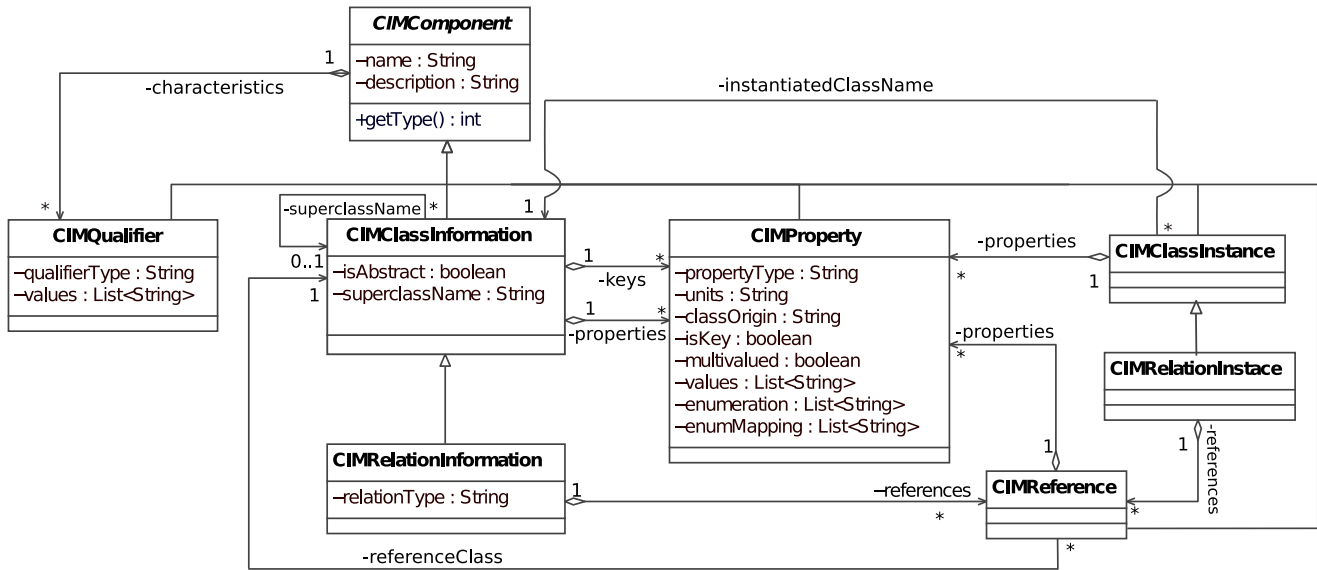


Figure 3. UML diagram of the internal CIM representation

rently running it or have reset capabilities. It is represented as a string using infix operators to make it more intuitive for users. At execution time, each operator is evaluated with the result of the recursive evaluation of its children as argument.

Search paths are evaluated by first finding all the instances at the end of the association that satisfy the given conditions on their properties, and then working backwards, navigating through the associations trying to find the origin instance. This method is used to avoid semantic-changing situations in 1-to-N associations that could arise when the search is done breadth-first from the origin node.

The resulting implementation allows arbitrary stacking of conditions on associated nodes, and arbitrarily large expressions based on these predicated nodes.

#### 4. Related Work

CIM has been used as information model to build different monitoring systems for Grid infrastructures. In Mao et al. [12] a monitoring system of web resources, RMCS, is presented. Experimental results have shown that the system provides scalable and flexible capabilities and satisfies the web application requirements. The work presented by Ravelomanana et al. [14] is another example of CIM-based grid monitoring system.

CIM is also used by Corcho et al. [4] to define S-OGSA (Semantic-OGSA), an extension of OGSA to support the explicit handling of semantics.

UDDI [17] (Universal Description Discovery and Integration framework) could be considered as an alternative to MDS for service discovery in OGSA-based Grids. In [2]

UDDI is evaluated as a provider of resource discovery services for Grids. The authors conclude that it is only appropriate for small Grids, since the overhead may be prohibitive for large Grids.

Garg et al. [10] present a service that offers a single API for both resources and services using the UDDI registry for service discovery and the MDS directory for resource discovery. It uses CIM as the basis for its resource ontology. Similarly, Tangmunarunkit et al. [16] use CIM to model another resource ontology and develop matchmaker functionality that is not limited to strict syntactic matching between client and server.

The Information Service of the middleware gLite [11] is R-GMA (Relational Grid Monitoring Architecture) [3, 15]. R-GMA is based on relational database and servlet technology for its implementation, and only supports SQL as a query language.

CIS [13] is another Information Service using CIM as the underlying information model. It can be deployed in UNICORE [7] Grids to obtain information about resources and services.

#### 5. Conclusions and Future Work

This work has presented mechanisms to integrate CIM information in the MDS4 repository and a query service that supports querying the CIM information using the CIM association model in full. The integration of the CIM model into MDS4 has many applications in the Grid area, since resource discovery and job scheduling benefit from the structured and interrelated information that an information

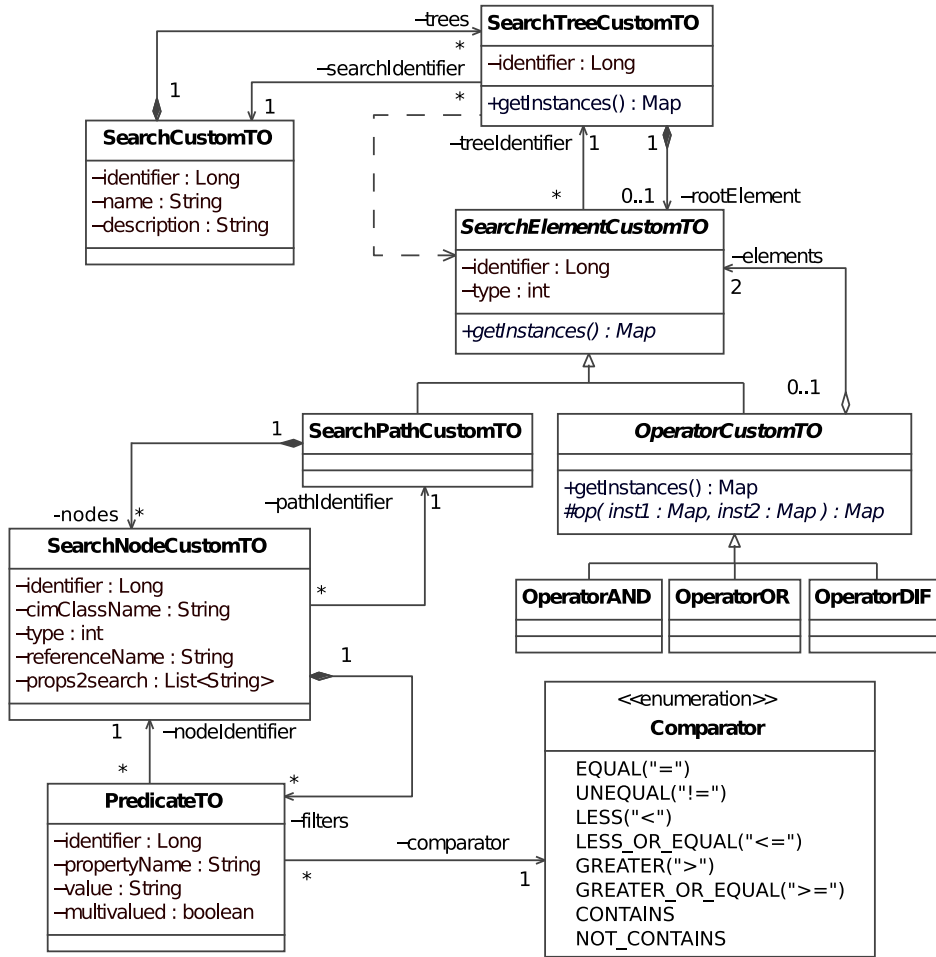


Figure 4. UML diagram of the query representation.

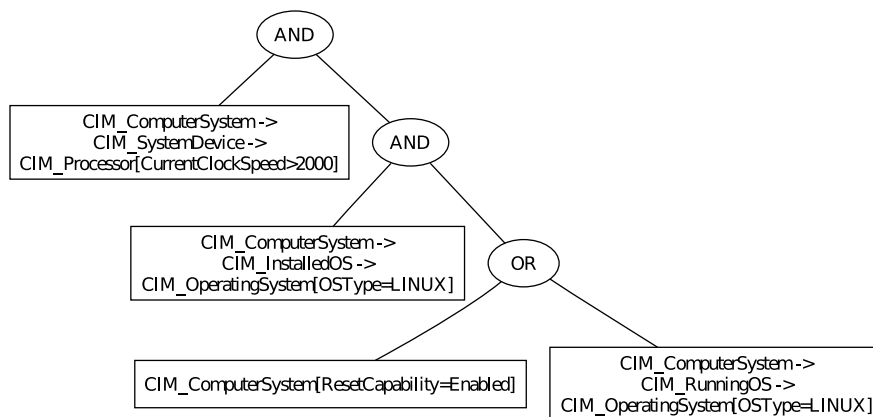


Figure 5. Example query tree.

model like CIM makes accessible. The query service also allows to specify complex expressions that would be impractical to formulate on other information models, like GLUE, without ad-hoc code. The infrastructure for supporting this integration and the query implementation has been the focus of this paper.

Unlike previous versions which used an LDAP repository, information in the MDS4 component of Globus is maintained in memory. This fact could limit the usefulness of our approach, although a change in the MDS4 persistence method or the use of other repository — preferably without changing the XML codification of data — could solve this problem.

Also, the internal representation of the CIM model is stored in memory. We plan to keep this representation in persistent storage in order to reduce memory overhead and allow to query large grids seamlessly. We are also working on XQuery back-end based and human language-like queries in other repositories.

## Acknowledgements

This work was funded by the Ministry of Education and Science of Spain under Project TIN2007-67537-C03, and by the Galician Government (Xunta de Galicia) under Project PGIDIT06PXIB105228PR.

## References

- [1] S. Andreozzi, M. Sgaravatto, and C. Vistoli. Sharing a conceptual model of grid resources and services. In *Proceedings of Computing in High Energy Physics, CHEP2003*, La Jolla, USA, March 2003.
- [2] E. Benson, G. Wasson, and M. Humphrey. Evaluation of UDDI as a provider of resource discovery services for OGSA-based grids. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium, IPDPS 2006*, page 9, Rhodes, Greece, April 2006.
- [3] A. W. Cooke, A. J. G. Gray, L. Ma, W. Nutt, J. Magowan, M. Oevers, P. Taylor, R. Byrom, L. Field, S. Hicks, J. Leake, M. Soni, A. J. Wilson, R. Cordenonsi, L. Cornwall, A. Djaoui, S. Fisher, N. Podhorszki, B. A. Coghlan, S. Kenny, and D. O'Callaghan. R-GMA: An information integration system for grid monitoring. In *Proceedings of The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 462–481, Catania, Italy, November 2003.
- [4] O. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, and C. Goble. An overview of S-OGSA: A reference semantic grid architecture. *Journal of Web Semantics*, 4(2):102–115, June 2006.
- [5] Distributed Management Task Force, Inc. DSP0201: Specification for the representation of CIM in XML. version 2.3.0c [online]. 2007. Available at: [http://www.dmtf.org/standards/published\\_documents/DSP0201\\_2.3.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0201_2.3.0.pdf) [Last accessed August 2008].
- [6] Distributed Management Task Force, Inc. Common Information Model (CIM) standards [online]. 2008. Available at: <http://www.dmtf.org/standards/cim> [Last accessed August 2008].
- [7] D. W. Erwin. UNICORE - a grid computing environment. *Concurrency and Computation: Practice and Experience*, 14(13–15):1395–1410, 2002.
- [8] I. Foster. Globus toolkit version 4: Software for service-oriented systems. *Journal of Computer Science and Technology*, 21(4):513–520, July 2006.
- [9] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration [online]. 2002. Available at: <http://www.globus.org/alliance/publications/papers/ogsa.pdf> [Last accessed August 2008].
- [10] S. Garg, F. Lee, S. Garg, and P. Mullan. Grid service discovery integrating UDDI and MDS. In *Proceedings of the 1st Asian Semantic Web Conference, ASWC 2006, Service Discovery on the WWW Workshop, sdisco '06*, Beijing, China, September 2006.
- [11] EGEE gLite middleware [online]. Available at: <http://glite.web.cern.ch/glite/> [Last accessed August 2008].
- [12] H. Mao, L. Huang, and M. Li. Web resource monitoring based on Common Information Model. In *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing, APSCC 2006*, pages 520–525, GuangZhou, China, December 2006.
- [13] A. S. Memon, M. S. Memon, P. Wieder, and B. Schuller. CIS: An information service based on the Common Information Model. In *Proceedings of the 3rd IEEE International Conference on e-Science and Grid Computing, eScience 2007*, pages 465–473, Bangalore, India, December 2007.
- [14] S. Ravelomanana, S. C. S. Bianchi, C. Joumaa, and M. Sibilla. A contextual GRID monitoring by a model driven approach. In *Proceedings of the Advanced International Conference on Telecommunications and Services, AICT-ICIW 2006*, page 37, Guadeloupe, French Caribbean, February 2006.
- [15] R-GMA: Relational grid monitoring architecture [online]. Available at: <http://www.r-gma.org/> [Last accessed August 2008].
- [16] H. Tangmunarunkit, S. Decker, and C. Kesselman. Ontology-based resource matching in the grid - the grid meets the semantic web. In *Proceedings of the SemanticWeb, ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 706–721, Sanibel, USA, October 2003.
- [17] Universal description, discovery and integration (UDDI) [online]. Available at: <http://uddi.xml.org/> [Last accessed August 2008].