WebDMF: A Web-based Management Framework for Distributed Services

George Oikonomou and Theodore Apostolopoulos

***Abstract*—This paper presents WebDMF, a Web-based Framework for the Management of Distributed services. It is based on the Web-based Enterprise Management (WBEM) family of standards and introduces a middleware layer of entities called "Representatives". WebDMF can be integrated with existing WBEM infrastructures and is not limited to monitoring. On the contrary, it is capable of actively modifying the run-time parameters of the managed application. Its design is abstract and suitable for a variety of distributed services, such as grids and content delivery networks. The paper includes a discussion on WebDMF's design, implementation and advantages. We also present experiments on an emulated network topology as an indication of the framework's viability.**

***Index Terms*—Common Information Model, Distributed Services Management, Web-based Enterprise Management, WebDMF**

I. Introduction

Legacy management approaches such as the Simple Network Management Protocol (SNMP) [1], target single nodes and are mainly used for the management of devices. The current paradigm of highly decentralized, distributed services presents new challenges and increased complexity in the area of network and systems management. There is need for solutions that are better-suited for software and services. Such solutions should also take into account the managed services' distributed nature.

We present the design, implementation and evaluation of WebDMF, a Web-based Management Framework for Distributed services. It uses standard web technologies: Its core is based on the Web-based Enterprise Management (WBEM) family of specifications [2], [3], [4]. It is not limited to monitoring but is also capable of modifying the run-time parameters of the managed service. Due to its abstract design it has wide scope and is suitable for the management of a variety of services. We have particularly studied its application for the management of grids [5], content delivery networks and web server load-balancing schemes.

The paper makes the following contributions:

- WebDMF can be integrated with existing WBEM infrastructures. We demonstrate how, in doing so, it can achieve its goal without need for modifications to the managed service.
- We provide indications for the viability of the approach through a preliminary performance evaluation.

In the context of this paper, the term "distributed" is used to describe a system, application or service that is hosted on multiple nodes interconnected over a network. Therefore this includes deployments such as distributed file systems, computer clusters and grids. However, multiprocessor, multi-core, distributed shared memory (DSM) and similar systems are considered out of the scope of our work, even though they are very often referred to as "distributed".

Section II briefly outlines existing efforts in the field of distributed systems management. In order to familiarize the reader with some basic concepts, section III introduces the WBEM family of standards. In section IV we describe WebDMF's architectural design. Implementation details and preliminary evaluation results are presented in section V. Finally, in section VI we discuss our conclusions.

II. Related Work

Existing research and development efforts investigate techniques for the management of distributed applications and services. The Open Grid Forum's GMA [6] and gLite's R-GMA [7] focus on monitoring grids. MonALISA [8] and the CODE toolkit [9] have wider scope but still only perform monitoring.

There are some proposals that can go beyond monitoring. The Unified Grid Management and Data Architecture (UGanDA) [10] contains an infrastructure manager called MAGI. MAGI has many features but is limited to the management of UGanDA deployments. MRF is a Multi-layer resource Reconfiguration Framework for grid computing [11]. It has been implemented on a grid-enabled Distributed Shared Memory (DSM) system called Teamster-G [12].

Policy-based management is another proposal. The Internet Engineering Task Force (IETF) and the Distributed Management Task Force (DMTF) are working together to develop new policies. Examples of policy management applied on distributed systems can be found in [13] and [14].

Finally, we should mention emerging web service-based management initiatives, such as OASIS' Web Services Distributed Management (WSDM) [15] and DMTF's Web Services for Management (WS-Man) [16]. Due to their importance they are further discussed in section VI.

Manuscript received March 22, 2008.
George Oikonomou is with the Athens University of Economics and Business, Department of Informatics, 76 Patission Str. 104 34 Athens Greece (phone: +30 210 8203155; fax +30 210 8203159; e-mail: geo@aueb.gr).
Theodore Apostolopoulos is with the Athens University of Economics and Business, Department of Informatics, 76 Patission Str. 104 34 Athens Greece (e-mail: tca@aueb.gr).

III. Web-based Enterprise Management

Web-Based Enterprise Management (WBEM) is a set of specifications published by the Distributed Management Task Force (DMTF). WBEM is made up of three core components. The "Common Information Model" (CIM) is a set of specifications for the modeling of management data [2]. It is an object-oriented, platform-independent model maintained by the DMTF. It includes a "core schema" with definitions that apply to all management areas. It also includes a set of "common models" that represent common management areas, such as networks, hardware, software and services. Finally, the CIM allows manufacturers to define technology-specific "extension schemas" that directly suit the management needs of their implementations.

WBEM adopts the client-server paradigm. For the interaction between WBEM entities (clients and managed elements), it uses a set of well-defined request and response data packets. CIM elements are encoded in XML in accordance with the xmlCIM specification [3]. The resulting XML document is then transmitted over a network as the payload of an HTTP message. This transport mechanism is called "CIM Operations over HTTP" [4]. The term CIM-XML is often used to refer to the combination of terms "CIM over HTTP" and xmlCIM.

A WBEM server is made up of components as portrayed in Fig. 1. The WBEM client does not have direct access to the managed resources. Instead, it sends requests to the CIM Object Manager (CIMOM), using CIM over HTTP. The CIMOM handles all communication with the client. It delegates requests to the appropriate providers and returns responses.

Providers act as plugins for the CIMOM. They are responsible for the actual implementation of the management operations for a managed resource. Therefore, providers are implementation-specific. The repository is the part of the WBEM server that stores the definitions of the core, common and extension CIM schemas.

A significant number of vendors have started releasing WBEM products. The SBLIM open source project offers a suite of WBEM-related tools. Furthermore, OpenPegasus, OpenWBEM and WBEMServices are some noteworthy, open source CIMOM implementations. There are also numerous commercial solutions.

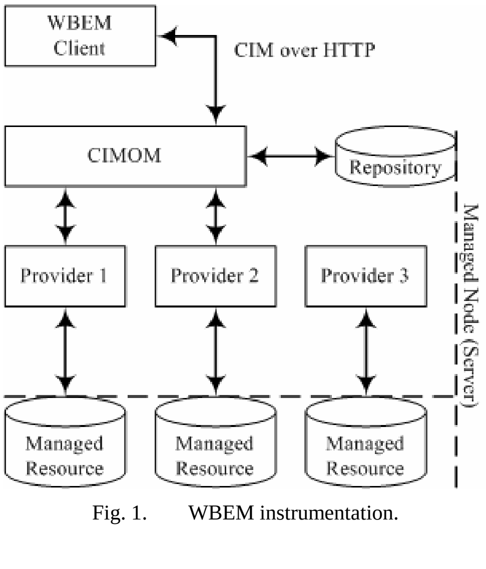

Fig. 1. WBEM instrumentation.

IV. WebDMF: A Web-based Management Framework for Distributed Services.

WebDMF stands for Web-based Distributed Management Framework. This treats a distributed system as a number of host nodes that are interconnected over a network and share resources to provide services to the end user. The proposed framework's aim is to provide management facilities for the nodes. Through their management, we achieve the management of the entire deployment.

The architecture is based on the WBEM family of technologies. Nodes function as WBEM entities; clients, servers or both, depending on their role in the deployment. Communication between nodes is performed in accordance with CIM-XML.

WebDMF's design introduces a middleware layer of entities that we call "Management Representatives". They act as peers and form a management overlay network. This new layer of nodes can be integrated with an existing WBEM-based management infrastructure. Representatives act as intermediaries between existing WBEM clients and CIM Object Managers. In our work, we use the terms "Management" and "Service" node when referring to those entities. Fig. 2 displays the three management entities mentioned above, forming a very simple topology.

This resembles the "Manager of Managers" (MoM) approach. However, in MoM there is no direct communication between domain managers. In WebDMF, representatives are aware of the existence of their peers. Therefore, it adopts the "Distributed Management" approach. By distributing management over several nodes throughout the network, we can increase reliability, robustness and performance, while network communication and computation costs decrease [17].

A. *Management and Service Nodes*

A "Management Node" is a typical WBEM client. It is used to monitor and configure the various operational parameters of the distributed service. Any existing WBEM client software can be used without modifications.

A "Service Node" is the term used when referring to any node – member of the distributed service. For instance, in the case of a computational grid, the term can describe an execution host. Similarly, in a content delivery network a service node can be an intermediate relay node or a node hosting content. As stated previously, the role of a node in a particular distributed deployment does not affect the design of our framework.

Typically, a Service Node executes an instance of the (distributed) managed service. As displayed in Fig. 3 (a), a WebDMF request is received by the CIMOM on the Service Node. A provider specifically written for the service handles the execution of the management operation. The existence of such a provider is a requirement. In other words, the distributed service must be manageable through WBEM. Alternatively, a service may be manageable through SNMP, as shown in Fig. 3 (b). In such a case the node may still participate in WebDMF deployments but some functional restrictions will apply.

ISBN:978-988-98671-9-5

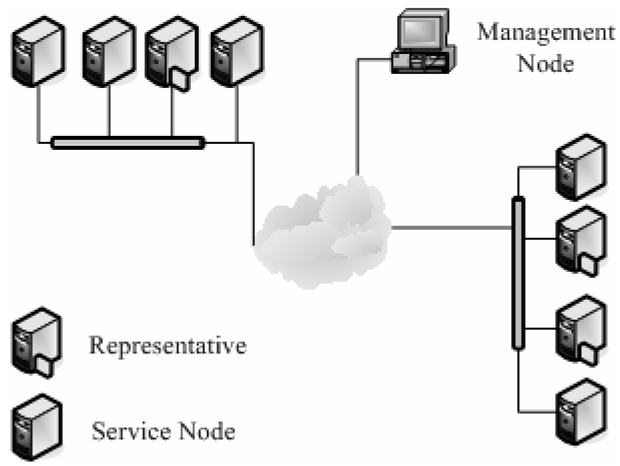

Fig. 2. Management entities.

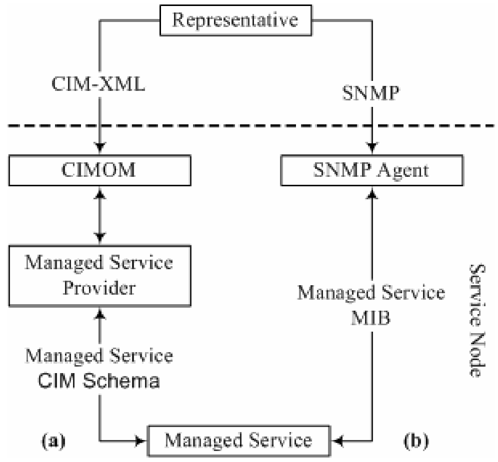

Fig. 3. Service node.

B. Management Representative

As stated previously, the framework introduces an entity called the "Management Representative". This entity initially receives a request from a WBEM client (management node) and performs management actions on the relevant service nodes. After a series of message exchanges, it will respond to the initial request. A representative is more than a simple 'proxy' that receives and forwards requests. It performs a number of other operations including the following:

- Exchanges messages with other representatives regarding the state of the system as a whole.
- Keeps a record of Service Nodes that participate in the deployment.
- Redirects requests to other representatives.

The initial requests do not state explicitly which service nodes are involved in the management task. The decision about the destination of the intermediate message exchange is part of the functionality implemented in the representative. The message exchange is transparent to the management node and the end user.

In order to achieve the above functionality, a representative is further split into building blocks, as shown in Fig. 4. It can act as a WBEM server as well as a client. Initial requests are received by the CIMOM on the representative. They are delegated to the WebDMF provider module for further processing. The module performs the following functions:

- Determines whether the request can be served locally.
- If the node can not directly serve the request then it selects the appropriate representative and forwards it.
- If the request can be served locally, the representative creates a list of service nodes that should be contacted and issues intermediate requests.
- It processes intermediate responses and generates the final response.
- Finally, it maintains information about the distributed system's topology.

In some situations, a service node does not support WBEM but is only manageable through SNMP. In this case, the representative attempts to perform the operation using SNMP methods. This is based on a set of WBEM to SNMP mapping rules. There are limitations since it is not possible to map all methods. However, even under limitations, the legacy service node can still participate in the deployment.

C. Domains

In a WebDMF deployment, a representative is responsible for the management of a group of service nodes, either on its own or in cooperation with other peers. We use the term "Domain" when referring to such groups. The relationship between domains and representatives is many-to-many. Thus, a representative may be responsible for the management of more than one domain.

Domains are organized in a hierarchical structure. The top level of the hierarchy (root node of the tree) corresponds to the entire deployment. The exact rationale behind the domain hierarchy of each individual deployment can be based on a variety of criteria. For example, a system may be separated into domains based on the geographical location of nodes or on the structure of an organization. In any case, the hierarchy can affect the performance, ease of management and scalability of the system. The domain hierarchical structure and the relationships between domains and nodes are depicted in the class diagram in Fig. 5.

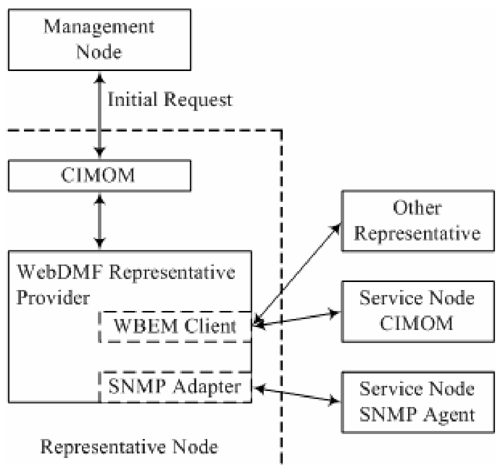

Fig. 4. WebDMF representative.

ISBN:978-988-98671-9-5

D. *CIM Schemas and Operations*

WebDMF defines two categories of management operations:

- Horizontal (Category A).
- Vertical (Category B).

Horizontal Operations enable management of the WebDMF overlay network itself. Those functions can, for example, be used to perform topology changes. The message exchange that takes place does not involve Service Nodes. Therefore, the managed service is not affected in any way. When a service node joins the network, it registers itself with a representative. This can be performed by a script on the service node itself or manually by a user. However, there is no automated discovery mechanism for new service nodes.

The registration operation is very lightweight. However, its "semi-manual" nature makes WebDMF more suitable for deployments with relevantly infrequent topology changes. It is less suitable for systems with very frequent and abrupt topology changes (disconnecting nodes, roaming nodes), such as wireless, ad-hoc networks.

Vertical operations read and modify the CIM schema on the Service Node, thus achieving management of the target application. Typical examples include:

- Setting new values on CIM objects of many service nodes.
- Reading operational parameters from service nodes and reporting an aggregate (e.g. sum or average).

In line with the above, we have designed two CIM Schemas for WebDMF, the core schema ("WebDMF_Core") and the request factory. They both reside on the representatives' repositories.

The former schema models the deployment's logical topology, as discussed earlier. The class diagram presented in Fig. 5 is the actual diagram representing WebDMF_Core. Horizontal functions correspond to WBEM operations on instances of classes declared in this schema.

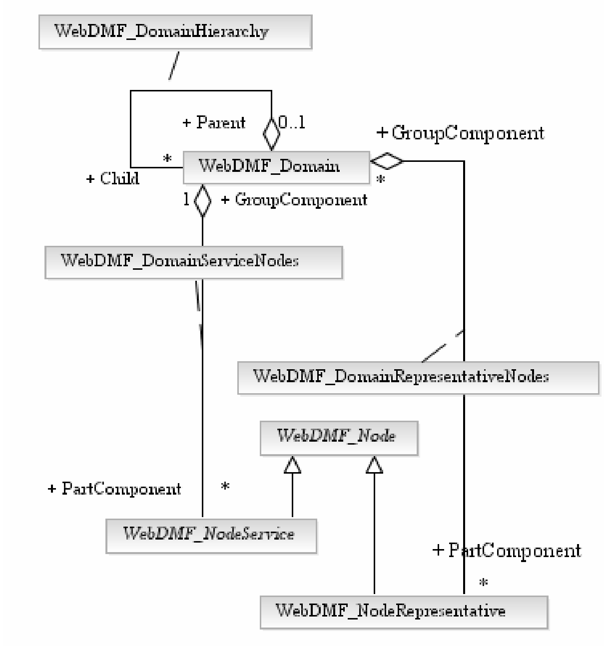

Fig. 5. Domains and nodes – The WebDMF core schema.

The latter schema corresponds to vertical functions. Users can call WBEM methods on instances of this schema. In doing so, they can define the management operations that they wish to perform on the target application. Each request towards the distributed deployment is treated as a managed resource itself. For example, a user can create a new request. They can execute it periodically and read the results. They can modify it, re-execute it and finally delete it.

In order to complete a vertical operation, the following message exchange takes place:

- The management node sends a `CreateInstance()` WBEM message to any representative. This requests the creation of a new instance for class `WebDMF_RequestWBEM`. This instance defines the management operation that needs to be performed on service nodes.
- The representative determines whether the request can be served locally. If not, it chooses the appropriate representative and issues a new `CreateInstance()` request.
- If the representative can serve the request, it generates a list of service nodes that must be contacted. This is based on the values of the "`DomainName`" and "`Deep`" properties of the newly generated instance.
- The representative sends the appropriate requests to service nodes. The type of CIM operation used for those requests is also based on values of properties in the instance. For example, this operation can be a `ModifyInstance()`.
- After all service nodes have been contacted, responses are sent. The instance on the first representative contains results. It remains available to the user for potential modification and/or re-execution. All other intermediate instances are deleted.

Fig. 6 displays the two most important CIM classes of the request factory. Class WebDMF_RequestFactory is abstract and common parent for all other classes declared in the schema. In the above example, it is inherited and instantiated by class WebDMF_RequestWBEM. The schema declares several more classes. Due to length restrictions, we do not present a detailed description of class properties and methods.

Request factory classes are generic. They are not related in any way with the CIM schema of the managed application. This makes WebDMF appropriate for the management of a wide variety of services. Furthermore, no re-configuration is needed when the target schema is modified.

V. Implementation and Performance Evaluation

A. *Some Implementation Details*

The WebDMF representative is implemented as a single shared object library file (.so). It is comprised of a set of WBEM providers. Each one of them implements CIM management operations for a class of the WebDMF schemas.

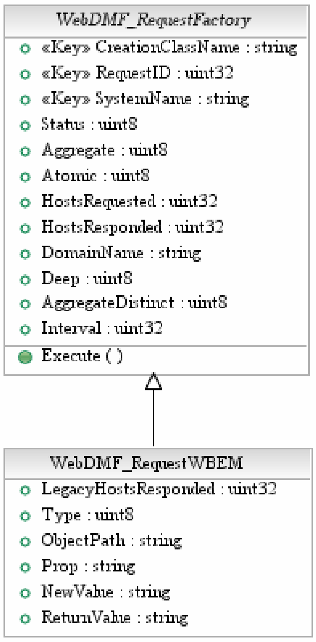

Fig. 6. Two classes of the request factory.

The interface between the CIMOM and providers complies with the Common Manageability Programming Interface (CMPI). Providers themselves are written using C++ coding. This does not break CIMOM independence, as described in [18].

The representative was developed on Linux 2.6.20 machines. We used gcc 4.1.2 and version 2.17.50 of binutils. Testing took place using version 2.7.0 of the Open Pegasus CIMOM.

B. Performance Evaluation

In order to evaluate WebDMF, we installed a testbed environment using ModelNet [19]. Obtained from actual code execution on an emulated network topology and used as an indication of the solution's viability, the results presented here are not simulation results. They also bring out one of WebDMF's key features, the ability to perform changes to the managed service.

The topology emulated by ModelNet represents a wide-area network. It consists of 250 virtual nodes situated in 3 LANs with each LAN having its own gateway to the WAN. The 3 gateways are interconnected via a backbone network, with high bandwidth, low delay links. We have also installed two WebDMF representatives (nodes R1 and R2).

In this scenario, the distributed managed service is a content delivery network implemented with the OpenCDN (oCDN) open source software [20]. Service nodes correspond to oCDN Origin nodes. Each of those registers with an oCDN centralized control entity called "Request Routing and Distribution Management" (RRDM). We have also designed a CIM schema and the relevant WBEM providers that enable management of OpenCDN nodes. Fig. 7 portrays the emulated topology and test scenario. For clarity reasons, we omit service nodes residing in other domains.

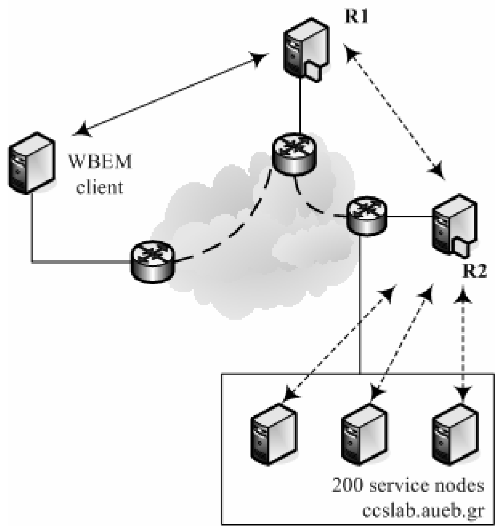

Fig. 7. Emulated topology and test scenario.

In our experiment, we wish to perform a change in the 200 service nodes residing in domain "ccslab.aueb.gr". To be more specific, we wish to set them to register with a different RRDM. This involves changing properties of the oCDN_Origin instance on the service node's CIMOM.

The client will form a WBEM `CreateInstance()` request for class `WebDMF_RequestWBEM` of the request factory. It is initially sent to the WebDMF Representative (R1). The request will get forwarded to R2. R2 will send `ModifyInstance()` requests to the 200 service nodes. R2 reports to R1. R1 sends the final response to the client.

The above experiment was repeated 200 times. Table I summarizes the results with times measured in seconds. Consider the fact that each repetition involves 204 request-response exchanges among various nodes. Furthermore, consider that packets crossing the network are of a small size (a few bytes). The total execution time includes the following:

- Communication delays during request-response exchanges. This includes TCP connection setup for all WBEM message exchanges.
- Processing overheads on R1 and R2. This is imposed by WebDMF's functionality.
- Processing at the service nodes to calculate the requested value and generate a response.

TABLE I. EVALUATION RESULTS.

Metrics		Values
Repetitions	N	200
Central Tendency	Arithmetic Mean	3.851944
	Median	3.869485
Dispersion	Variance	0.005281
	Standard Deviation	0.072670
Quartiles	Q0 (min)	3.710042
	Q1	3.797917
	Q3	3.910481
	Q4 (max)	3.997615
95% Confidence Interval for the Mean		
	From	3.841873
	To	3.862016

The absolute value of the average completion time may seem rather high. However, in general terms, processing times are minimal compared to TCP connection setup and message exchange. With that in mind, we can see that each of the 204 request-responses completes in 18.88 milliseconds on average. This is normal.

After 200 repetitions we observe low statistical dispersion (variance and standard deviation). This indicates that the measured values are not widely spread around the mean. We draw the same conclusion by estimating a 95% confidence interval for the mean. This indicates that the same experiment will complete in the same time under similar network load conditions.

VI. CONCLUSIONS

Table II compares WebDMF with some of the solutions outlined in section II. For this comparison we consider two factors, i) The solution's ability to go past simple monitoring and ii) the product's scope; whether it is generic or focuses on specific distributed systems.

Existing monitoring solutions are generic. However, ones that offer more features and the ability to "set" tend to be more focused. We wanted to design a framework that would be generic enough and suitable for a wide variety of services. At the same time, it should not be limited to monitoring. WebDMF achieves that by detaching the details of the managed service from the representative logic. Management functions for a specific service are implemented by WBEM providers on the service nodes. Representatives unify those on a deployment scale.

WebDMF has some other noteworthy advantages:

- It is based on WBEM. This is a family of open standards. WBEM has been considered adequate for the management of applications, as opposed to other approaches (e.g. SNMP) that focus on the management of devices.
- It provides interoperability with existing WBEM-based management infrastructures without need for modifications.

WebDMF is resource-centric, something that may seem to be a step in the opposite direction compared to emerging web service-based initiatives. However, those approaches are model-agnostic. They do not define properties and operations for the managed resources [15]. Further study of [21] and [22] shows that WebDMF can act in the resource-layer of a service-based management deployment.

TABLE II. COMPARING WEBDMF WITH OTHER SOLUTIONS.

Name	Set	Scope
OGF's GMA		Wide
gLite – R-GMA		Focused
CODE		Wide
UGanDA – MAGI	Y	Focused
MRF – Teamster-G	Y	Focused
MonALISA		Wide
WebDMF	Y	Wide

REFERENCES

[1] W. Stallings, *SNMP, SNMPv2, SNMPv3, RMON 1 and 2*. Addison Wesley, 1999.
[2] *CIM Infrastructure Specification*, DMTF Standard DSP0004, 2005.
[3] *Representation of CIM in XML*, DMTF Standard DSP0201, 2007.
[4] *CIM Operations over HTTP*, DMTF Standard DSP0200, 2007.
[5] G. Oikonomou, and T. Apostolopoulos, "Using a Web-based Framework to Manage Grid Deployments," in *Proc. The 2008 International Conference on Grid Computing and Applications (ICPDC 08)*, to be published.
[6] *A Grid Monitoring Architecture*, Open grid Forum GFD.7, 2002.
[7] A. W. Cooke, et al, "The Relational Grid Monitoring Architecture: Mediating Information about the Grid," *Journal of Grid Computing*, vol. 2, no. 4, pp. 323-339, 2004.
[8] I.C. Legrand, H.B. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, M. Toarta, and C. Dobre, "MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications," in *Proc. Computing in High Energy and Nuclear Physics (CHEP)*, Interlaken, Switzerland, 2004.
[9] W. Smith, "A System for Monitoring and Management of Computational Grids," in *Proc. International Conference on Parallel Processing (ICPP'02)*, 2002, p. 55.
[10] K. Gor, D. Ra, S. Ali, L. Alves, N. Arurkar, I. Gupta, A. Chakrabarti, A. Sharma, and S. Sengupta, "Scalable enterprise level workflow and infrastructure management in a grid computing environment," in *Proc. Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05)*, Cardiff, UK, 2005, pp. 661–667.
[11] P.-C. Chen, J.-B. Chang, T.-Y. Liang, C.-K. Shieh, and Y.-C. Zhuang, "A multi-layer resource reconfiguration framework for grid computing," in *Proc. 4th international workshop on middleware for grid computing (MGC'06)*, Melbourne, Australia, 2006, p. 13.
[12] T.-Y. Liang, C.-Y. Wu, J.-B. Chang, and C.-K. Shieh, "Teamster-G: a grid-enabled software DSM system," in *Proc. Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05)*, Cardiff, UK, 2005, pp. 905–912.
[13] N. Damianou, N. Dulay, E. Lupu, M. Sloman, "Ponder: A Language for Specifying Security and Management Policies for Distributed Systems. The Language Specification - Version 2.3". Research Report DoC 2000/1, Imperial College of Science Technology and Medicine, Department of Computing, London. [online] Available: http://www-dse.doc.ic.ac.uk/Research/policies/ponder/PonderSpec.pdf
[14] D. C. Verma, S. Calo, and K. Amiri, "Policy-based management of content distribution networks," *IEEE Network*, vol. 16, iss. 2, pp. 34–39, 2002.
[15] *An Introduction to WSDM*, OASIS committee draft, 2006.
[16] *Web Services for Management (WS Management)*, DMTF Preliminary Standard DSP0226, 2006.
[17] M. Kahani and P. H. W. Beadle, "Decentralised approaches for network management," *ACM SIGCOMM Computer Communication Review*, vol. 27, iss. 3, pp. 36–47, 1997.
[18] *Common Manageability Programming Interface*, The Open Group, C061, 2006.
[19] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostic, J. Chase, and D. Becker "Scalability and Accuracy in a Large-Scale Network Emulator," in *Proc. 5th Symposium on Operating Systems Design and Implementation (OSDI)*, December 2002.
[20] OpenCDN Project [Online]. Available: http://labtel.ing.uniroma1.it/opencdn/
[21] *WS-CIM Mapping Specification*, DMTF Preliminary Standard DSP0230, 2006.
[22] *WS-Management CIM Binding Specification*, DMTF Preliminary Standard DSP0227, 2006.