Modeling Storage System Performance for Data Management in Cloud Environment using Ontology

Stanisław Polak[1], Darin Nikolow[1], Renata Słota[1] and Jacek Kitowski[1,2]

[1] AGH - University of Science and Technology, Institute of Computer Science, Mickiewicza 30, 30-059 Kraków, Poland

[2] ACC Cyfronet AGH, Nawojki 11, 30-950 Kraków, Poland

Abstract. The progress made in the field of Cloud computing and the continuously growing users demand for services with guaranteed storage performance parameters bring new challenges. The storage system monitoring, resource scheduling and performance prediction are essential for successful operation of the given distributed environment and for fulfillment of the Service Level Agreement. Taking into account the heterogeneity of storage resources in distributed environments it is essential to provide a transparency of monitored storage system performance parameters. In this paper we present a common storage system model regarding Quality of Service requirements and dynamics of performance parameters. We also present the process of the storage ontology development based on this model, and we show an use-case of the proposed ontology in a storage monitoring service.

1 Introduction

As the Cloud computing paradigm gains popularity new challenges for the service providers arise. Users have a set of requirements, described more or less formally in the Service Level Agreement (SLA), regarding the service they demand. A subset of these requirements may concern Quality of Service (QoS) of the storage systems where the users keep their data, especially if the users are interested in running data oriented applications, for example, on the Grid. The storage system monitoring, resource scheduling and performance prediction are essential for successful operation of the given distributed environment and for fulfillment of the SLA.

Modern data centers provide the infrastructure necessary for constructing the mentioned distributed environment [1]. Various services like SaaS, PaaS, etc. can be present in this environment sharing the infrastructure. Since data sets are often replicated in such environments for high availability or performance reasons, those services could optimize their access to the data using storage performance prediction. Taking into account the heterogeneity of storage resources in distributed environments it is essential to provide a transparency of the monitored storage system performance parameters necessary for the prediction.

Using semantics for QoS description of available services allows for better service selection [2]. Similarly, using ontologies to describe storage resources is expected to

Polak S., Nikolow D., Slota R. and Kitowski J..
Modeling Storage System Performance for Data Management in Cloud Environment using Ontology.
DOI: 10.5220/0003352100540063
In *Proceedings of the International Workshop on Semantic Interoperability* (IWSI-2011), pages 54-63
ISBN: 978-989-8425-43-0

bring more efficient storage resource usage and better storage service interoperability in distributed environment with QoS/SLA requirements.

In this paper we present the OntoStor ontology and its development method. The OntoStor ontology describes Mass Storage Systems (MSS) like: Hierarchical Storage Management (HSM) systems, disk arrays and local storage based MSS (local disks attached to a server), and their QoS aspects. Our develepment method involves: 1) Designing the Common Mass Storage System Model (CMSSM), 2) Creating its standards-based version called CIM based Common Mass Storage System Model (C2SM), 3) Developing the ontology itself.

The rest of the paper is organized as follows: first, state of the art on modeling of storage systems is presented. In section 3 the CMSSM and the C2SM models are presented. Next, the proposed ontologies of storage resources are described. Application of one of them is shown in section 5. The last section summarizes the paper.

2 State of the Art

There are a few popular models of information systems like: CIM, SMI-S, and GLUE. All of them are shortly characterized below.

2.1 Common Information Model

Common Information Model (CIM) [3] is an open standard developed by the Distributed Management Task Force (DMTF). It is a hierarchical, object oriented model of management elements in Information Technology (IT) environment. CIM as a general model, is not bound to a particular implementation. It consists of two parts: an Infrastructure Specification and a Schema.

CIM Infrastructure Specification contains a description of object oriented meta-model based on Unified Modeling Language (UML), i.e., Meta Schema, details for integration with other models, and a grammar of Managed Object Format (MOF) language. The basic elements of Meta Schema are: Schemas, Classes, Properties, and Methods. Additional elements are: Indications, Associations, and References.

CIM Schema contains a set of predefined classes, their properties, methods, and dependencies among these classes. CIM Schema consists of two separated layers: Core Model and Common Models.

Core Model defines common classes (the basic dictionary), which are used by elements of Common Models

Common Models have a set of predefined models. These models are independent of any implementation or technology, and describe particular areas of management.

CIM is the main component of Web-Based Enterprise Management (WBEM) systems used for a distributed management of computing environments [4].

CIM has many schemas relevant to MSS: Storage Devices, Storage Services, Storage Capabilities and Settings, Storage Statistics, Physical Component. Because the HSM systems as well as most of the performance related parameter of MSS are not directly represented in CIM we decided to build our model - C2SM.

Existing CIM schemas can be extended, and new ones can be designed. In extending schemas, CIM specification recommends using only classes defined in the Core model.

2.2 Storage Management Initiative - Specification

Storage Management Initiative - Specification (SMI-S) [5] is a CIM based standard formulated by Storage Networking Industry Association (SNIA), which defines an interface to manage heterogeneous data storage environment consisting of: data storage devices, data storage systems, and management applications. Nowadays more than 500 products are SMI-S compliant. In SMI-S, CIM classes are grouped in profiles (e.g. disk array), and each of them can have subprofiles. To be SMI-S compliant, the product vendor has to implement all the mandatory CIM classes specified in the profile. Subprofiles describe a part of management domain, and represent optional functionality of product (e.g. client can discover a remote management interface).

In SMI-S 1.1 which is ANSI standard, there are four groups of profiles: Storage, Host, Fabric Topology, and Server. The first one has profiles, which directly regards to data storage devices but they do not represent the needed performance parameters.

2.3 Grid Laboratory Uniform Environment

Grid Laboratory Uniform Environment (GLUE) [6] is a conceptual, object-oriented, information model of Grid environments, and its main aim is to provide interoperability among elements of the Grid infrastructure.

The basic elements of the model are main entities, which represent the core concepts of the Grid environment: resource, service, location, etc. Conceptual models of computing and storage services are defined based on the main entities. These entities are described with UML Class Diagrams. GLUE does not allow adding new classes or associations to the model. Two approaches are available for extending the information associated with the existing classes: placing additional information in a special attribute, which is present in each class or creation of key/value pairs and then linking them to the class.

In GLUE the model of MSS is defined with the following classes: StorageService, StorageServiceCapacity, StorageAccessProtocol, StorageEndpoint, StorageShare, StorageShareCapacity, StorageManager, DataStore, ComputingService. In order to have dynamics represented in the model it should be extended but GLUE is less extensible than CIM, is Grid oriented, and GLUE developers plan to use CIM for the modeling of GLUE 2 [7].

The models mentioned above do not fully address the performance aspects of MSS which are needed for the proper management of storage in distributed environment with QoS constraints. This fact motivated us to develop a CIM based model - C2SM.

3 Models of Data Storage Systems

In this section we present our two models describing MSS. The first one is the Common Mass Storage System Model (CMSSM), and the second one, the CIM based Common mass storage System Model (C2SM), is a modified version of CMSSM arisen as a result of taking into account the CIM standard. Due to the use of the C2SM model as a base for the OntoStor ontology, we describe it in more detail.

3.1 The CMSSM Model

The CMSSM model describes three kinds of MSS: the HSM systems, disk array systems, and local disk systems. This model consists of sets of parameters describing the current state of a specific MSS, its configuration (e.g. total capacity of MSS, maximal read transfer rate), and the parameters of physical devices and media (e.g. number of tape tapes, tape block size). CMSSM uses our earlier work on modeling HSM systems presented in [8]. The current version of CMSSM can be found in [9].

3.2 The C2SM Model

The CIM based version of CMSSM — the C2SM model consists of a set of classes (see Fig. 1) briefly described further. The `AGH_StorageSystem` class is the main class and it stores common information for all kinds of MSS. The `AGH_DiskArraySystem`, `AGH_LocalDiskSystem` and `AGH_HSMSystem` classes represent, the mentioned above, kinds of MSS.

The `AGH_DiskArraySystem` contains information about disk array systems. These systems consist of: a disk array (the `CIM_ComputerSystem` class) and a server (the `CIM_ComputerSystem` class).

The `AGH_LocalDiskSystem` class stores information about local disk systems. These systems comprise a server (the `CIM_ComputerSystem` class) having hard disks.

The `AGH_HSMSystem` contains information common for the HSM systems. These systems include: a server (the `CIM_ComputerSystem` class), media libraries (the `AGH_MediaLibrary` class), and disks (the `CIM_DiskPartition` class). Media libraries contain: slots (the `AGH_Slot` class), changer devices (the `AGH_ChangerDevice` class), and drives (the `AGH_MediaAccessDevice` class). The `AGH_HSMMedia`, `AGH_HSMDriveState`, `AGH_HSMFile`, and `CIM_LogicalFile` classes describe the states of media, drives, and files stored in HSM respectively.

4 The OntoStor Ontology

OntoStor [10] is a research project with the purpose of developing an ontology-based methodology concerning the organization of data access in Grid environments. Different kinds of MSS and services for monitoring and estimation of data access are semantically described. The semantic description allows for efficient use of MSS and for easier creating and integration of new Grid-enabled, data access applications.

Based on the C2SM model described above, the OntoStor ontology has been created (See Fig. 2). At the beginning, this model was written in Managed Object Format (MOF), and then converted into Web Ontology Language (OWL) format using the 'cim2owl' tool [11]. Next the result file was modified by a human, e.g.: redundant

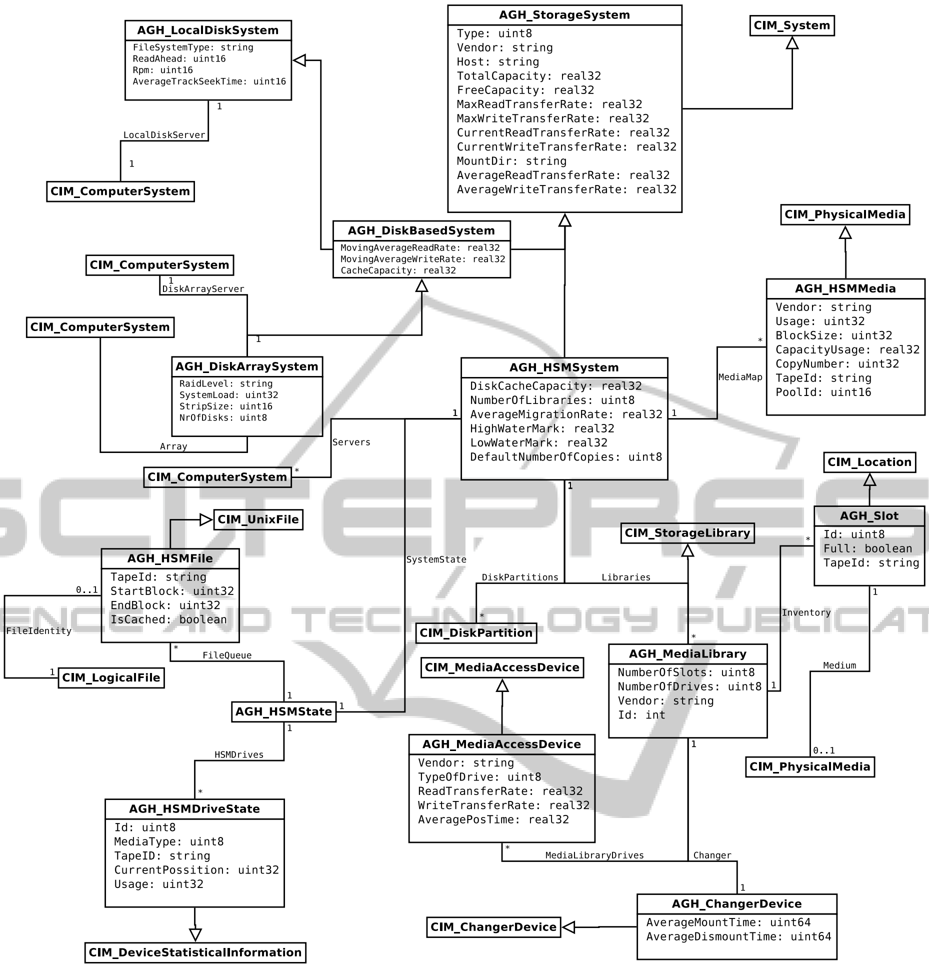

Fig. 1. The C2SM model.

components (individuals, properties, classes) were removed, closure axioms for some classes were added, new properties, and individuals representing existing MSS systems were created. As a result we received an ontology in which the classes of the C2SM model were represented by OWL classes, and the class properties by datatype properties in OWL. For example, the 'BlockSize' property of the `AGH_HSMMedia` class is represented in OWL as follows:

```
<owl:DatatypeProperty rdf:ID="AGH_HSMMedia__BlockSize">
  <rdfs:range rdf:resource="http://www.w3.org/2001/
                          XMLSchema#nonNegativeInteger"/>
  <rdfs:domain rdf:resource="#AGH_HSMMedia"/>
</owl:DatatypeProperty>
```

The three kinds of MSS mentioned in section 1 are represented by the following classes: `AGH_HSMSystem`, `AGH_DiskArraySystem` and `AGH_LocalDiskSystem`.

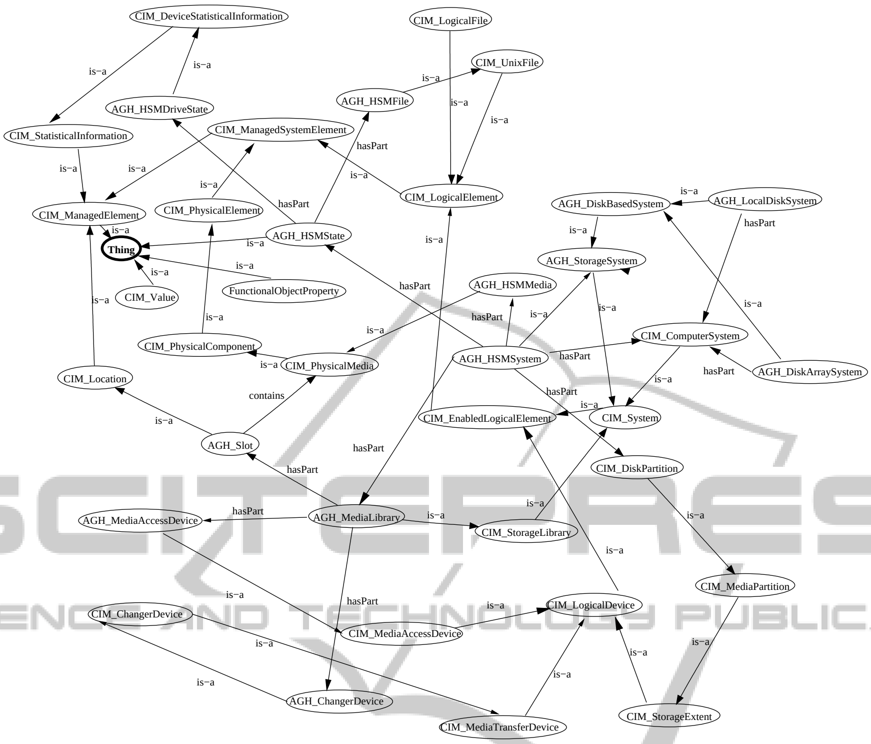

Fig. 2. The OntoStor ontology.

Their components are represented by the following classes: `CIM_ComputerSystem`, `AGH_MediaLibrary`, `AGH_HSMMedia`, and `CIM_DiskPartition`. Components of media libraries of the HSM system are represented by the `AGH_ChangerDevice` class, the `AGH_MediaAccessDevice` class, and the `AGH_Slot` class. The `AGH_HSMMedia`, `AGH_HSMDriveState`, `AGH_HSMFile`, and `CIM_LogicalFile` classes represents information about states of media, drives, and files stored in HSM.

Using this ontology we described the resources in our testing environment, i.e., the HP 660ex magneto-optical library and the ATL 7100 tape library, and their components. Below we present a fragment of an OWL file containing the description of the mentioned above, magneto-optical library.

```
<owl:Thing rdf:about="#Magneto_Optical_library_HP_660ex">
  <rdf:type rdf:resource="#AGH_MediaLibrary"/>
  <AGH_MediaLibrary__NumberOfSlots
    rdf:datatype="&xsd;unsignedByte">
      128
   </AGH_MediaLibrary__NumberOfSlots>
  <AGH_MediaLibrary__NumberOfDrives
    rdf:datatype="&xsd;unsignedByte">
      4
  </AGH_MediaLibrary__NumberOfDrives>
```

```
  <AGH_MediaLibrary__VendorString
    rdf:datatype="&xsd;string">
      HP
   </AGH_MediaLibrary__VendorString>
  <hasPart rdf:resource="#Drive_HP_5200"/>
  <hasPart rdf:resource="#Slot_1_HP"/>
  <hasPart rdf:resource="#Slot_2_HP"/>
</owl:Thing>
```

As we can see this ontology, contains semantically described information about a concrete value of a concrete parameter of MSS. Thanks to this ontology we are able to find a kind of MSS and their components, base on values like numbers or strings, e.g., ”find media libraries which have four or less drives”.

4.1 The OntoStor-ATN Ontology

The ontology described above does not cover all applications areas, e.g., it can not be used to identify a kind of MSS based on the names of the attributes. Since this kind of functionality was needed in a another project, an alternative version of the ontology — the OntoStor-ATN (Attribute Name) ontology was created (see Fig. 3). This ontology defines two main concepts:

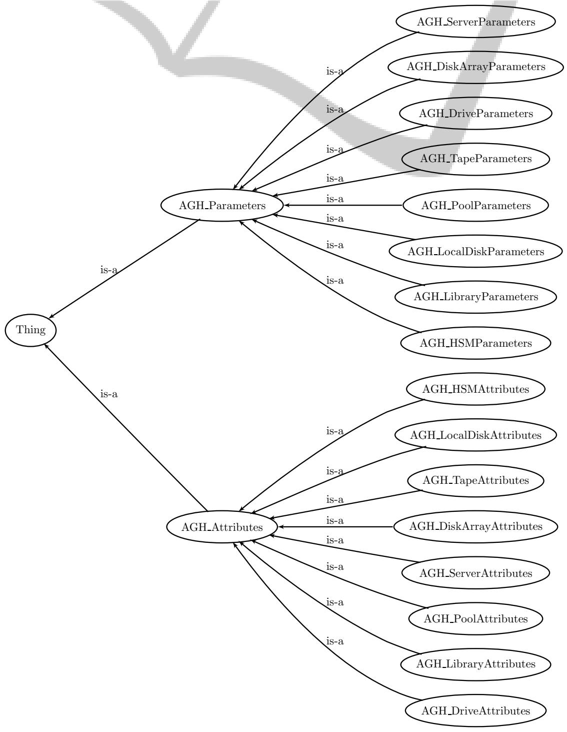

Fig. 3. The OntoStor-ATN ontology.

AGH_Parameters — basic class for concepts representing resources. i.e. MSS and their components

AGH_Attributes — basic class for concepts representing attributes of resources

Subclasses of `AGH_Parameters`, like e.g.: the `AGH_ServerParameters` class, describe concepts, which can be identified with concrete resources, e.g., server. This ontology does not contain individuals of these concepts. Subclasses of the second main class, i.e., `AGH_Attributes`, define concepts of resource attributes of individual resources, e.g., the `AGH_DiskArrayAttributes` concept represents attributes of disk arrays. In this case, individuals of the `AGH_*Attributes` classes are defined in the ontology, e.g., mentioned above the 'BlockSize' property of the `AGH_HSMMedia` class is represent as the 'BlockSize' individual of the OWL `AGH_TapeAttributes` class. All individuals are associated with concepts of resources by the 'hasValue' restriction — below we show how the mentioned 'BlockSize' individual is assigned to the `AGH_TapeParameters` class representing tapes.

```
<owl:Class rdf:about="AGH_TapeParameters">
  <rdfs:subClassOf rdf:resource="AGH_Parameters"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAttribute"/>
      <owl:hasValue rdf:resource="BlockSize"/>
     </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

In the same way we assign other individuals, in result this ontology contains complete set of concepts related with MSS and individuals representing their attributes.

5 Application of the OntoStor-ATN Ontology

The OntoStor-ATN ontology is used in the PL-Grid project [12]. The objective of this project is to provide Polish scientists a Grid-based high performance computing environment enabling doing e-science research.

One of the studies undertaken on PL-Grid is data management for Virtual Organization (VO). In Grid environments, resource providing on the basis of VO is particularly justified, because Grid applications have often high requirements in relation to hardware resources. These requirements can refer to computational power and data storage resources. For Data Oriented Applications, which are executed on Grid, the computational power is not the only requirement — providing storage resources guaranteeing the Quality of Service (QoS) is also necessary. In heterogeneous environments, like Grid, prediction of the performance of shared resources is a hard task. In order to meet the QoS requirements two kinds of information are needed: information about the current storage resource performance utilization and information about the scheduled data transfer. To obtain this information a monitoring system taking into account the heterogeneity of MSS should be used. This system has to be configured automatically and independent of the MSS being monitored and should provide a unified monitoring layer for the QoS related parameters.

In [13] a system using the CMSSM model and the OntoStor-ATN ontology is described. General QoS aspects of storage resources are described by different metrics. For different kinds of resources, these metrics have different meaning, e.g., total capacity of a local disk can be received directly from their parameters, while the total capacity of the HSM system has to be calculated as the sum of the disk cache capacity and capacity of each medium.

The OntoStor-ATN ontology is used to describe storage resources and their attributes; QoS capabilities of these resources are described by enriched version of the QoSOnt ontology [14]. Because the resources are semantically described, it is possible to identify the current kind of resources, based on the name of the attributes, and as a result, the calculation of a concrete metric is also possible.

Within the PLGrid project a semantic-oriented monitoring system called the FiVO SLAM (Service Level Agreement) monitoring system [15] has been implemented. It is a part of the FiVO framework [16] – a system for deployment and negotiation of dynamic VO.

6 Summary

In this paper we described the process of developing the OntoStor ontology, allowing to describe more precisely MSS and providing performance related properties. Since this ontology does not cover all application areas, the OntoStor-ATN ontology has been created. This ontology was used in a semantic-oriented monitoring system. The lesson learned is that the internal ontology structure is essential in the case of representing QoS/SLA parameters. It is important for the monitoring applications to have the parameter names represented as classes.

The ontologies presented in the paper are based on the proposed CIM based model - C2SM, which describes storage resources and their performance parameters. By using our model and ontologies we achieved transparency of monitored storage system performance parameters and therefore the interoperability of monitoring and estimation services is possible.

In the future we plan to extend our model and ontologies by adding a new kind of MSS based on disk pools.

Acknowledgements

This research is partially supported by the MNiSW grant nr N N516 405535 and AGH-UST grant nr 11.11.120.865.

References

1. Kant, K.: Data center evolution. Computer Networks 53 (2009) 2939–2965
2. Dobson, G.: Towards unified QoS/SLA ontologies. In: Proceedings of Third International Workshop on Semantic and Dynamic Web Processes (SDWP). (2006)
3. Common Information Model (CIM) Standards: http://www.dmtf.org/standards/cim/ (last access 15 Nov, 2010)

4. Web-Based Enterprise Management (WBEM): http://www.dmtf.org/standards/wbem (last access 15 Nov, 2010)
5. Storage Management Initiative Specification (SMI-S): http:// www.snia.org/tech_activities/standards/curr_standards/smi (last access 15 Nov, 2010)
6. Grid Laboratory Uniform Environment (GLUE): http://forge.gridforum.org/sf/projects/glue-wg. (last access 15 Nov, 2010)
7. Andreozzi, S., Burke, S., Field, L., Konya, B.: Towards GLUE 2: evolution of the computing element information model . Journal of Physics: Conference Series 119 (6) (2008) http://iopscience.iop.org/1742-6596/119/6/062009.
8. Nikolow, D., Słota, R., Kitowski, J.: Grid services for HSM systems monitoring. In Wyrzykowski, R., Dongarra, J., Waśniewski, J., eds.: Proceedings of 7-th International Conference, PPAM 2007, Gdańsk, Poland, September 2007. Volume 4967 of LNCS., Springer (2008) 321–330
9. Polak, S.: The OntoStor project report no. 4.5.1-v2 (2010) http://www.icsr.agh.edu.pl/trac/ontostor/browser/CMSSM/raport.odt (in Polish).
10. The OntoStor project: http://www.icsr.agh.edu.pl/ontostor/ (last access 15 Nov, 2010)
11. The CIM2OWL tool: http://fivo.cyf-kr.edu.pl:18888/space/CIM2OWL (last access 15 Nov, 2010)
12. The PL-Grid project: http://www.plgrid.pl/en (last access 15 Nov, 2010)
13. Słota, R., Nikolow, D., Skałkowski, K., Kitowski, J.: Data management with quality of service in PL-Grid environment. In: KU KDM 2010 : third ACC Cyfronet AGH users' conference : Zakopane March 18–19, 2010 : proceedings, ACK Cyfronet (2010) 35–46
14. Dobson, G., Lock, R., Sommerville, I.: Qosont: a qos ontology for service-centric systems. In: 31st EUROMICRO Conference on Software Engineering and Advanced Applications. (2005) 80–87
15. Funika, W., Kryza, B., Słota, R., Kitowski, J., Skałkowski, K., Sendor, J., Król, D.: Monitoring of SLA Parameters within VO for the SOA Paradigm. In Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J., eds.: Proceedings of Parallel Processing and Applied Mathematics - PPAM 2009, 8th International Conference, Wroclaw, Poland, September 2009. Volume II of LNCS., Springer (2010) 115–124
16. Kryza, B., Dutka, L., Słota, R., Kitowski, J.: Security Focused Dynamic Virtual Organizations in the Grid based on Contracts. In Cunningham, P., Cunningham, M., eds.: Collaboration and the Knowledge Economy, Issuses, Applications, Case Studies. Volume 5, II., IOS Press (2008) 1153–1160