

An Interoperable Grid Information System for Integrated Resource Monitoring Based on Virtual Organizations

Timo Baur · Rebecca Breu · Tibor Kálmán · Tobias Lindinger · Anne Milbert ·
Gevorg Poghosyan · Helmut Reiser · Mathilde Romberg

Received: 4 November 2008 / Accepted: 17 August 2009 / Published online: 3 September 2009
© Springer Science + Business Media B.V. 2009

Abstract In many Grid infrastructures different kinds of information services are in use, which utilize different incompatible data structures and interfaces to encode and provide their data. Homo-

geneous monitoring of these infrastructures with the monitoring data being accessible everywhere independently of the middleware which provided it, is the basis for a consistent status reporting on the Grids' resources and services. Thus, interoperability or interoperation between the different information services in a heterogeneous Grid infrastructure is required. Monitoring data must contain the identity of the affected Virtual Organization (VO) so that it can be related to the resources and services the VO has allocated to enable VO-specific information provision. This paper describes a distributed architecture for an interoperable information service, which combines data unification and categorization with policies for VO membership, VO resource management and data transformations. This service builds the basis for an integrated and interoperating monitoring of Grids, which provide their data to more than one VO and utilize heterogeneous information services.

This work has been partly funded by the D-Grid Initiative of the German Federal Ministry of Education and Research (FKZ01IG07010A).

T. Baur · H. Reiser
Leibniz Supercomputing Centre, Boltzmannstr. 1,
85748 Garching, Germany

R. Breu · M. Romberg (✉)
Jülich Supercomputing Centre,
Forschungszentrum Jülich GmbH,
52425 Jülich, Germany
e-mail: m.romberg@fz-juelich.de

T. Kálmán
Gesellschaft fuer wissenschaftliche Datenverarbeitung
mbH, Am Fassberg 11, 37077 Göttingen, Germany

T. Lindinger
Ludwig-Maximilians-Universität München,
Oettingenstr. 67, 80538 München, Germany

A. Milbert
Max Planck Institute for Gravitational Physics,
Am Mühlenberg 1, 14476 Golm, Germany

G. Poghosyan
Steinbuch Centre for Computing,
Forschungszentrum Karlsruhe GmbH,
Hermann-von-Helmholtz-Platz 1,
76344 Eggenstein-Leopoldshafen, Germany

Keywords Grid monitoring · Virtual Organizations · Grid interoperability · Grid information system · VO-specific information provisioning

1 Introduction

With Grid computing becoming mature and being used in large production infrastructures [1–5], the

role of monitoring is of growing importance for their successful operation and usage. Grid monitoring observes the status of a Grid infrastructure, in particular its resources, services and processes including user jobs, and thereby facilitates a Grid's reliable operation and a high availability of its components. It also builds the basis for checking and controlling the compliance of a Grid's service quality with the service level agreements (SLAs) made between providers and Virtual Organizations (VOs). It enables service benchmarking, correct accounting and billing as well as scheduling of resource and service usage.

To fulfill such assignments, Grid monitoring relies on data provided by information services. In spite of the importance of monitoring, large infrastructures with stakeholders from different organizations often use incompatible technical realizations, apply heterogeneous data structures to encode their data, and offer different interfaces to provide it. This is caused either by the absence of standardization, or, where standards exist, by designs and best-practice approaches neglecting the existing standards. The situation is further complicated by the fact that many standards are still evolving and that, repeatedly, transitions to new versions must be incorporated into the infrastructures. As a result, the provisioning of high-quality Grid operations requires solutions for interoperability and integration of Grid information services and data. Monitoring data needs to be accessible in the Grid independently of the organization or the technical platform which provides the access.

VO-specific access control thereby plays an important role. Consumers of monitoring data—respectively VOs—need the data specifically related to the resources and services they operate or which have been allocated to them. The data gathered by the Grid's information services should therefore be provisioned to a user according to his organizational context, that is the VOs he is a member of. We call such a mechanism *VO-awareness*.

In the following, we describe a distributed monitoring architecture for an interoperable and integrated information service as developed within the German e-Science monitoring project D-MON [6] as well as its implementation within the D-Grid infrastructure set up by the German

e-Science initiative [7]. The developed service combines data unification and categorization with policies for VO membership, VO resource management, and data transformations between different data models. It realizes VO-aware access to monitoring data gathered from multiple resource providing organizations as well as different Grid middlewares.

The remainder of this paper is structured in the following way: Section 2 outlines related work by introducing architectures, standards and interfaces used for Grid monitoring as well as approaches to the integration problem and VO-based data provisioning. Section 3 then illustrates the practical relevance of the problem using the example of the D-Grid infrastructure, a heterogeneous Grid infrastructure which utilizes multiple middlewares and their respective monitoring services. Motivated and guided by this scenario, Section 4 presents requirements for a homogeneous Grid monitoring system which is capable of overcoming the identified issues. The design of the D-MON monitoring system is the topic of Section 5 and the details about the implementation are covered in Section 6. Results, first experiences as well as an outlook are given in Section 7.

2 Related Work

There are a multitude of concepts in the Grid community which handle the problem of monitoring Grid resources and services. As a basic blueprint, the Open Grid Forum has defined the Grid Monitoring Architecture (GMA, [8]), which describes major components for a Grid monitoring system as well as their essential interactions. The GMA is a producer/consumer-based architecture separating data discovery from data transfer. It uses directory services to support discovery and information delivery between information producers and consumers. A producer first contacts the directory services and registers its existence and the type of information it wants to publish to the Grid. A consumer uses the directory services to discover information of interest and locate the respective producers. The consumer then may initiate direct interaction with the discovered producer, and the producer in turn sends the

requested data to the consumer without further involvement of the directory service. GMA is scalable and avoids single points of failure.

Based on the GMA architecture, sophisticated architectures have been developed, such as the Aggregator Framework of the Monitoring and Discovering System (MDS4, [9]) or the Relational Grid Monitoring Architecture (R-GMA, [10]). MDS4 can be queried using the Web Services Resource Framework (WS-RF, [11]), while R-GMA acts like one large relational database utilizing standard SQL commands. Both architectures have their advantages and their implementations are important parts of the middleware implementations of the Globus Toolkit [12] and gLite [13].

To achieve interoperability, using the method of defining and adopting common open standards [14] and architectures is a common approach, which relies heavily on standardization and implementation processes. Because comprehensive implementations and roll-outs spanning different Grid communities are difficult, costly and often politically charged, in certain scenarios a coupling of the architectures is a reasonable alternative.

In such scenarios, where it is favorable to operate multiple different implementations, e.g. different middlewares, in parallel or where there is no agreement on or knowledge of a common standard, non-intrusive components, e.g. bridges, become necessary to enable an integration. System integration has been a topic for a long time. For example, [15] describes three kinds of architectural bridges which can be applied: a multi-architectural platform, management gateways and multi-architectural agents. In our work, we concentrated on realizing a management gateway which may act as a bridge between different monitoring services.

Hegering et al. [15] also point out that integration can be realized by bridging communication (e.g. interfaces), information (e.g. data description schema), organization (e.g. roles), and functional models (e.g. queries). For achieving successful data exchange, we concentrated on building a bridge which realizes compatibility of information and communication models in an agile way. This bridge furthermore enables the retrieval of monitoring data according to roles and organizations.

In the area of information models the Grid Laboratory for a Uniform Environment (GLUE) information schema [16] is an example of an abstract data model for describing Grid services and resources. The GLUE project originally was a joint effort by the European DataGrid (EDG, [17]) and DataTAG [18] projects and the international Virtual Data Grid Laboratory [19] project. Today many major Grid projects such as Enabling Grids for E-science [2], Worldwide Large Hadron Collider (LHC) Computing Grid [20] and the Open Science Grid [21] have adopted the GLUE schema. The GLUE Working Group in the Open Grid Forum (OGF, [22]) is working on improving the schema. It proposed a major revision of the GLUE schema, version 2.0. Thus there is a high possibility that it will be deployed in many infrastructures in parallel with other schemata. GLUE 2.0 is the only model for Grid resources that supports Virtual Organizations, called User-Domains in the schema. Other information models are for example Usage Record (UR, [23]) from OGF and the D-Grid Resource Description Language (D-GRDL, [24]). OGF's UR model with its interface Resource Usage Service (RUS, [25]) focuses on job accounting information and does not provide general resource information which are essential for Grid Monitoring. D-GRDL has been developed within the German e-Science initiative for describing arbitrary sets of resources. D-GRDL offers a flexible but very abstract framework for the description of monitoring data. An additional disadvantage of a data schema based on D-GRDL is that it would be specific to D-Grid.

A standardized interface for the exchange of state data between Grid components was proposed by the OGF, the Web Services Resource Framework (WS-RF, [11]), which is already widely used. Nevertheless, there are different versions in use and many popular monitoring services just ignore that standard. Other methods to exchange Grid monitoring data are using XML or REST (Representational State Transfer, cf. [26]).

As opposed to past approaches to integration of distributed systems, Grid computing incorporates the concept of Virtual Organizations. For this reason, the necessity of VO-aware Grid services has been described in [27], and functional mechanisms as well as an architecture for interoperable and

VO-aware monitoring of dynamic Virtual Organizations have been discussed in [28]. In the work at hand, we apply and extend these ideas and detail the design and concrete implementation of an integrated, VO-aware service that provides Grid monitoring data.

3 Scenario

The D-Grid infrastructure set up by the German e-Science initiative [7] enables users from many scientific fields, grouped into communities realized as VOs, to exploit Grid computing for their specific applications. It is a large and complex Grid infrastructure with stakeholders from different organizations. In this environment incompatible technical realizations with heterogeneous data structures and interfaces emerge easily. The highly diverse sets of applications coming from the scientific communities are differently suited for the various available middleware solutions. Therefore, the infrastructure combines the three middleware installations of Globus Toolkit [12], gLite [13], and UNICORE [29] as well as dCache [30] and OGSA-DAI [31] for data management. In the D-Grid scenario, compute resources are offered through all three middleware solutions or through either of them. The communities use the middleware they are familiar with and which are best suited for their applications.

The communities constitute Virtual Organizations (VOs) which get access to the subset of resources contributed to that VO. In that sense a VO is not just a group of users but also consists of virtual resources and services, those which are available for use by the specific community. As a consequence, all Grid services have to be VO-aware to allow different contexts of resource and service allocation with respect to community specific requirements and technology.

Monitoring such a complex infrastructure is an ambitious task as each of the middleware implementations has its own notion of, and tools for monitoring. Up to now, monitoring for the different middleware solutions is often done in a middleware specific way: Globus sites use the MDS4 [32] information service with a GridCSM Web interface [27]; gLite resources are monitored

by Site Functional Test (SFT) [33] which bases on information from the BDII information service. The reports are displayed through a Web interface; UNICORE 5 sites are not monitored but UNICORE 6 sites are monitored through the Common Information Service (CIS) [34] that comes with a Google-maps user interface.

The result is a multitude of sources each providing only a fraction of the desired information. This complicates the operation of comprehensive Grid information services, which should be usable from everywhere in the Grid. Obviously, in that situation a more homogeneous and comprehensive approach is desirable.

Most of the Grid information services in use have the disadvantage of being focused on physical entities thus ignoring the actual mapping of resources and services onto VOs. A VO-centric approach would simplify a user's life as only information belonging to their VO would be extracted and presented to them, whereas all information they have no use of (such as the providers' cluster status) would be filtered. In addition, a VO-based privacy protection is possible. Analyses of the D-Grid scenario and concepts with respect to monitoring (cf. [35–37]) revealed several issues: One of the major shortcomings of a big infrastructure is that building several autonomous (monitoring) service components may result in several logical Grids without data interchange. To tackle this issue for the monitoring service, the German e-Science initiative established the research project D-MON [6].

The loss of interoperability in a Grid of middlewares may lead to problems in the Grid's operation, as the following use-case illustrates. For example, a scheduler used in one middleware is unaware of resource allocation mappings and jobs that belong to foreign middleware components. Parts of a Grid job may rely on resources available in a gLite based infrastructure such as EGEE [2], while other parts need resources available in a UNICORE based infrastructure such as DEISA [1]. Besides interoperation of authorization systems, job submission, data transfer, and scheduling services, also the monitoring systems have to interoperate in order to allow a smooth overall job scheduling as well as resource and service operation and maintenance.

Integrated monitoring services, which dynamically provide exhaustive information about the actual state of components from multiple monitoring systems, are a step towards integrated and interoperating Grids.

4 Requirements

The D-Grid scenario as described in Section 3 provides heterogeneous middleware installations in an infrastructure used by a large number of communities or Virtual Organizations. The monitoring of such an infrastructure must serve different purposes. In general, it should provide the information for Grid operation, for scheduling and accounting, for users searching and keeping track of resources, and for job monitoring. Within D-Grid these functions must integrate the deployed middleware solutions to provide the overall information. Thus an integrated gateway must have the ability to extract and transform monitoring data from different Grid information services in a non-intrusive way. In addition it must provide a VO-aware Grid service in a standardized way.

Since Globus Toolkit, gLite and UNICORE are installed on almost every compute resource in the Grid, a union of the monitoring data may generate duplicate as well as diverging data sets belonging to the same resource. An integrated monitoring system must augment data sets with provenance information. Amongst others the source and the time of retrieval of the data must be supplied. In this paper, provenance information is used to refer to the information that can help trace back the derivation history of the monitoring data. A detailed view of data provenance research in the scientific and business domains and a taxonomy of data provenance techniques has been surveyed in [38].

Access to monitoring data should be provided on a need-to-know basis and the different roles of the requesters have to be accounted for. For example, a Grid operator needs to know whether a system and its services are fully functional while a user in search of a resource must be supplied with resource characteristics like number of CPUs and available applications. As the collected information from all resources within an infrastruc-

ture spans all Virtual Organizations using the infrastructure and providing resources to it, the information sent to a requester must be VO-specific. Therefore the integration of VO management systems into Grid monitoring architectures is essential to obtain VO-specific information. Examples for VO-specific information are general information about a VO, the users belonging to the VO, the resources and services allocated to it, and which resources can be accessed by VO members. In order to implement VO-based monitoring proper access control has to be offered with it.

Other requirements are related to more general aspects. A common goal for all developments within the D-Grid environment is their scalability and extensibility. The monitoring system must be capable of providing information about the monitored items close to real time even when the numbers of sites, resources, users, roles, and VOs are increasing. Important for sustainability of the system is also the support of standards for protocols and data structures. The system to be developed should therefore be compliant with the Grid Monitoring Architecture (GMA) [8], the Web Services Resource Framework (WS-RF) [11], OGSA Data Architecture [39], and the emerging GLUE 2.0 schema [16] while security aspects such as authentication and authorization have to be taken into account. It is necessary to be able to pass monitoring information from the integrated monitoring platform back to middleware specific implementations, which is extremely important for Grid scheduling systems in multi-middleware Grids. They have to be capable to distribute jobs within the whole Grid based on information about the workload generated by all available middleware installations, not only a single one.

5 System Design

This section discusses design options for an integrated monitoring system fulfilling the requirements given in Section 4. An architecture for the integration of several heterogeneous monitoring systems builds the core of the system to be designed. On top of it interfaces to monitoring data and VO management information are modeled.

5.1 Basic Architecture

Several approaches to realize a monitoring system with the characteristics given above have been considered and their individual characteristics evaluated (see also [40]). To realize a data exchange between different monitoring systems, one option is building multiple bidirectional gateways between each pair of monitoring systems. This solution is hardly scalable as the overall number of gateways needed equals the squared amount of monitoring systems. Besides, preventing data duplication, which results from different naming and conventions, is a major challenge and the provision of all data in real-time can hardly be guaranteed.

An alternative would be defining one of the monitoring systems as the privileged system to collect all monitoring information. Then for each monitoring system only one gateway is needed which transfers data to and from the privileged system. While this solution reduces the amount of gateways needed to the linear amount of existing monitoring systems, it is limited to the capabilities of the privileged system and strongly depends on the future development of the privileged middleware; it furthermore contradicts the demand of strong interoperability. Additionally, it does not solve the problem of real-time data provision and duplicated data either.

The disadvantages of these options lead to consider another approach: An autonomous, middleware-independent service which utilizes a storage component, e.g. a database, with gateways connected to the underlying native monitoring services. This method would scale with the number of supported information providers, while being independent of a privileged middleware.

The comparison of the three design options identifies the third one as the preferable base architecture. The following subsections build on top of it.

5.2 Transformation of Monitoring Information

The interoperable Grid information system to be developed must gather and incorporate data from heterogeneous Grid monitoring systems. To transform the complex monitoring data into

an integrated data structure efficiently, an automated ETL-process (Extract–Transform–Load) is defined. ETL is a data integration technique which was first described in [41].

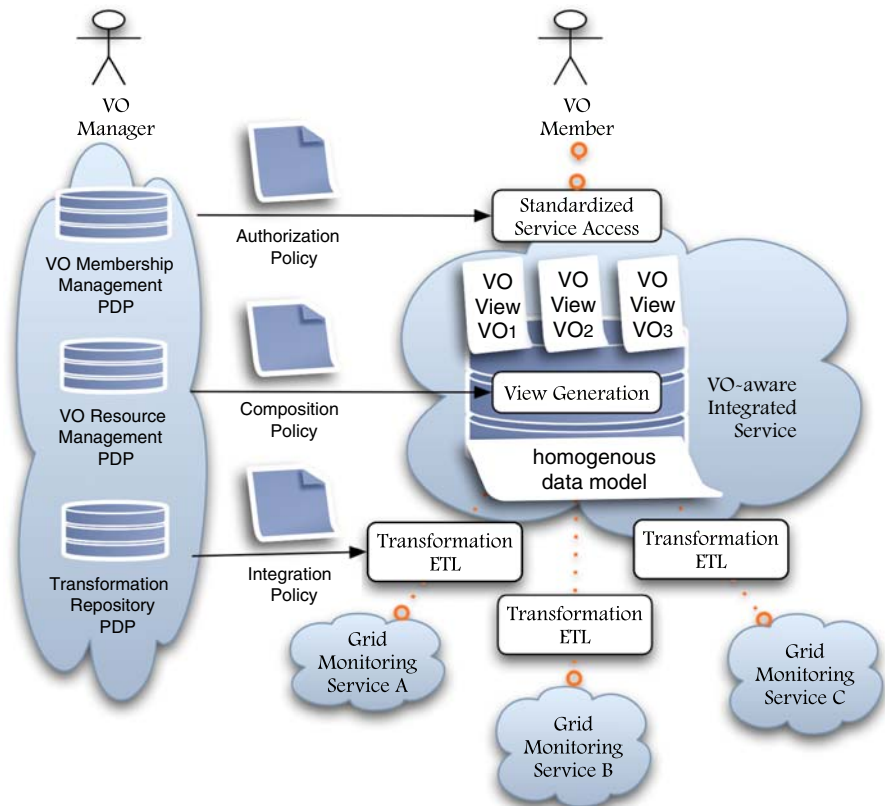
The monitoring information is collected asynchronously by Grid middleware specific extractors (E), using different protocols and data models. The extracted Grid monitoring data is parsed, checked and transformed (T) in a flexible way into a common data structure, e.g. the GLUE 2.0 schema. Finally the Grid monitoring data will be stored (L) in the target repository.

5.3 VO-aware Monitoring

In this subsection the design depicted so far is extended to cover the VO aspect described in Section 4. A VO-aware monitoring service can be realized by using external policy decision points (PDP) or policy engines which have knowledge about the actual mapping of resources and members onto VOs [27]. The term PDP has two flavors, organizational and technical. A political or organizational PDP denotes a committee which aggregates and coordinates the multilateral relationships of the different contractors in a Grid such as resource and service providers or customers organized as VOs. The organizational PDP may be hidden so that the contractual or political mappings are not explicitly known. A technical PDP defines the mappings in a technically and deterministically processable representation. Unlike the organizational PDP it must not be hidden. The representation can then be used as a template to compose the VOs as well as related monitoring data according to the VO's actual allocations of resources and services. In current Grid infrastructures, amongst others, services like the VO Membership Services (VOMS) and Grid Resource Repository Service (GRRS) [7] may serve as PDPs.

Figure 1 shows the functional components and transactions of the system described above. On the left hand side the components necessary for the integration of VO management layers into Grid monitoring architectures are shown. These are policy decision points and information bases, which define the rules for VO memberships, the resource composition of a VO, and the rules

Fig. 1 Layout of the integrated and VO-aware Grid monitoring service



for data transformations. The components at the bottom represent the various Grid monitoring services which provide the monitoring data. *Transformation/ETL* denotes the adaptors, which translate the data using the *integration policy*. The transformed data is then integrated into the *homogeneous data model* located in the storage at the *VO-aware integrated service*. At the center of the *VO-aware integrated service* the *View Generation* component is shown. It generates VO-related views of the data directed by the *composition policy*. Access to these views is granted by the service only to members of the corresponding VO, according to the *authorization policy* defined by a *VO Membership Management* system.

5.3.1 VO-aware View Generation

The *View Generation* component shown in Fig. 1 has to group the integrated monitoring data according to VOs. The data usually consists of information about resources, services, and jobs that exist within the Grid infrastructure. The VO re-

source management PDP provides the relation between resources or services and VOs. Converting this mapping into syntactically and semantically fitting data structures and integrating it with the homogeneous data model lays the basis for VO-based monitoring of resources and services. The identifier of resources and services recorded in the resource management system as well as in the source monitoring services must be unique and identical in order to allow a deterministic mapping of data sets to VO views. Updates in the VO resource management system must furthermore trigger new view generation cycles to keep the views consistent. For example the creation of a new VO must trigger the creation of an adapted access policy containing the new VO's permissions.

5.3.2 Access Authorization and Authentication

In order to regulate access to the monitoring data in a VO-aware way, authentication and authorization methods are necessary. For each query the

monitoring system needs to know the identity of the querying entity (user or service) that wants to retrieve the data. With this information, the VO membership management PDP can be asked for the entity's authorization. The authorization information contains the querying entity's VOs and thus the VOs' views can be determined which it is allowed to see. By using additional parameters, the user can constrain the query to specific VO views and a corresponding result can be generated.

Authentication and authorization methods can be based on VO management systems such as VOMS (Virtual Organization Management Service). They define the membership of accessing entities to specific VOs, information which can be used to find out which VO's data an accessing entity is authorized to retrieve. Newer methods use a combination of VOMS services and SAML (Security Assertion Markup Language) [42] to get the credentials of an accessing entity. As soon as credentials of a querying entity are known, the query can be filtered and the corresponding data sets can be provided.

5.3.3 Standardized Access Interface, Command Line Client and Web Client

For the client side a standardized access interface can simplify the addition of higher level interfaces like Web clients or command line clients. According to the requirements given in Section 4 the standardized database access interface must be compliant with the OGSA-DAI specification. For the web client a packet of portlets based on the GridSphere framework is chosen.

5.4 Distributed Setup

An overall architecture should account for scalability and efficiency, which includes minimal resource requirements and short access time to current information. The selected core architecture is an autonomous, middleware-independent service which utilizes a storage component with gateways connected to the underlying native monitoring services. There are three options to set up the storage component: central, clustered or federated.

A monitoring service with a central storage has to contain the integrated data from all connected site information services. This setup will not scale in large-scale scenarios, because a frequent update rate induced by a large number of gateways and sites as well as many client queries will cause high load in the central component.

An alternative is to use clustered storage systems distributed among the connected sites. This architecture is well known from database clusters but it has the disadvantage of being too unstable when distributed over the Internet.

A federated setup connects multiple autonomous services as federated services. The autonomous services are located directly at the resource providers' sites and query only the monitoring systems of the local provider. The federated services realize an overlay network which distributes the data and allows a further optimization by using aligned data distribution algorithms and protocols.

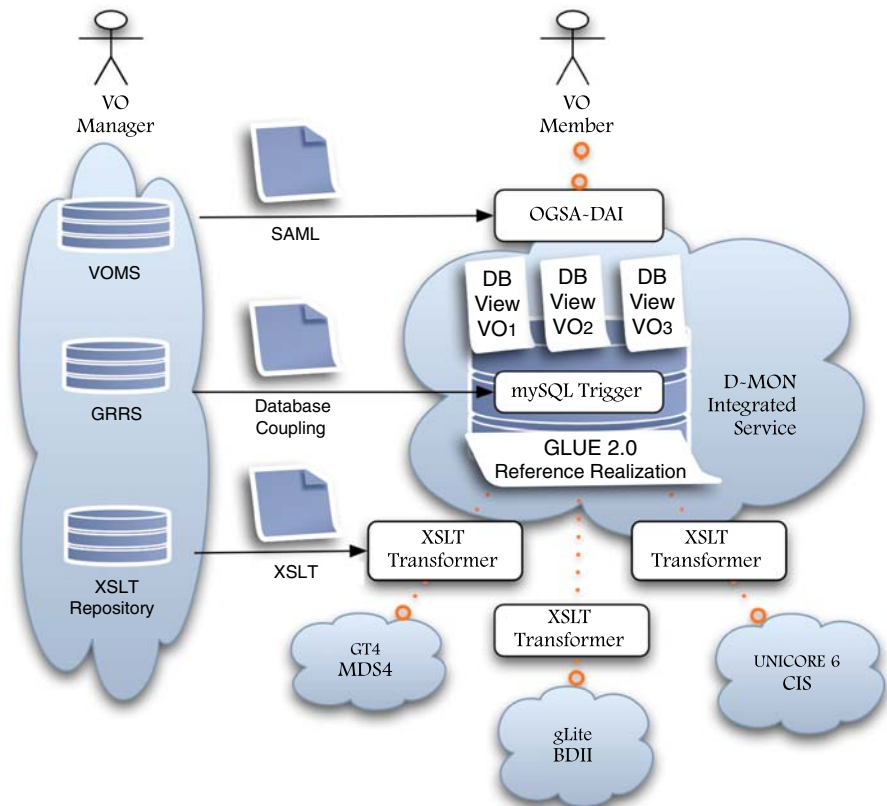
6 Implementation

The integrated monitoring system designed in Section 5 consists of an initially central storage incorporating a homogeneous data model, gateways to transform data models, policies to guide decisions, and a standardized access interface. In the following the implementation details of these components are given.

6.1 Integrated System

As discussed in Section 5, an autonomous component for storing all the monitoring data extracted by the gateways is needed. A database seems to fit the requirements best; it provides advanced filtering mechanisms and, based on relational database views, VO-specific monitoring views are easy to realize. Furthermore all modern database implementations can be operated in clustered or federated setups in a transparent way. Since in D-Grid all the information of VO management systems is stored in databases as well, using a database for the autonomous component has the advantage of being able to join the existing VO information with the data of the monitoring services. As a data

Fig. 2 Implementation of the integrated service D-MON



model the latest draft of the GLUE 2.0 schema is used together with the already existing SQL reference implementation [43]. Figure 2 depicts the system as implemented.

6.2 XSL Transformations for Schema Interoperability

The interface between middleware-specific information services and the D-MON information database is implemented as a three step process: extract, transform, and load. This process of extracting data from BDII, CIS, and MDS4, the three existing monitoring systems in D-Grid, transforming this data to the GLUE 2.0 schema, and loading it into the target database is realized by middleware-specific gateways, one per middleware. The extractor components of the gateways operate with the standard client APIs of the middlewares. The data transformation is implemented using Extensible Stylesheet Language Transformations (XSLT), which is a popular language for

processing XML data or transforming XML documents into other formats. This choice presented itself as most of the information services provide their data in XML format. The data is augmented with the identifiers of the information provider component to keep track of data provenance. In the following key aspects of the individual gateways' implementations are highlighted.

CIS, the information system of the UNICORE 6 middleware, already uses the GLUE 2.0 schema internally. Thus the XSL Transformations for the UNICORE gateway merely checks the XML retrieved from the CIS and maps it to the respective SQL-Commands. No special rearrangement or interpretation of the data is needed.

The information system of the gLite middleware, BDII, is based on the Lightweight Directory Access Protocol (LDAP) and provides output in LDAP Data Interchange Format (LDIF). The BDII gateway extracts the monitoring information using the Directory Services Markup Language (DSML), which is an XML representation

of the directory service information and can be easily transformed using XSL Transformations. Listing 1 illustrates an example of the XSL Transformations of the BDII gateway: The lines up to 30 parse the XML file. In the subsequent lines the respective SQL query is formed for the AdminDomain component of the GLUE 2.0 schema. For the integration of data provenance

information special labels are defined in lines 10–12 to identify the data source. An XSL template is defined in line 20 and it is applied to the matching DSML elements and their child nodes. Whenever new monitoring information is inserted into the database, a unique identifier is used. The lines 27–43 show how the identifier for the AdminDomain component is parsed. If monitoring data with the

```

1 <xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3  xmlns:dsml="http://www.dsml.org/DSML">
  <!-- Using the W3C XSLT and DSML namespaces -->
5
7  <!-- Creating text based output -->
  <xsl:output method="text" encoding="ISO-8859-1" />
9
  <!-- Labels to identify the data source -->
  <xsl:variable name="informationProvider">BDII</xsl:variable>
11 <xsl:variable name="sourceAddress">iwrdmon-bdii.fzk.de</xsl:variable>
  <xsl:variable name="insertTime">CURRENT_TIMESTAMP</xsl:variable>
13
  <!-- The input XML is a valid DSML -->
15 <xsl:template match="/">
  <xsl:apply-templates select="//dsml:dsml/dsml:directory-entries/dsml:entry"/>
17 </xsl:template>
19
  <!-- A template for the DSML entries -->
  <xsl:template match="//dsml:dsml/dsml:directory-entries/dsml:entry">
21   <xsl:variable name="DistinguishedName"><xsl:value-of select="@dn"/></
     xsl:variable>
  <xsl:choose>
23   <!-- Trying to find the GlueSiteUniqueID -->
     <xsl:when test="starts-with(@dn, 'GlueSiteUniqueID ')">
25
27     <!-- Parsing the unique ID of the AdminDomain -->
     <xsl:variable name="DistinguishedNameLocal">
     <xsl:value-of select="substring-after(substring-before(
29       $DistinguishedName, ',mds-vo-name=local,o=grid'), 'o-name=')"/>
     </xsl:variable>
31
     <!-- Creating SQL output by parsing the attributes -->
33     INSERT INTO GLUE20.AdminDomain (
       ID, Name, Description, Distributed,
35       informationProvider, sourceAddr, insertTime)
     VALUES (
37       "<xsl:value-of select="$DistinguishedNameLocal"/>",
       "<xsl:value-of select="dsml:attr[@name='GlueSiteName']/dsml:value"/>",
       "<xsl:value-of select="dsml:attr[@name='GlueSiteDescription']/
39         dsml:value"/>
       -<xsl:value-of select="dsml:attr[@name='GlueSiteOtherInfo']/
         dsml:value"/>",
41       "0",
       "<xsl:value-of select="$informationProvider"/>",
43       "<xsl:value-of select="$sourceAddress"/>",
       <xsl:value-of select="$insertTime"/>)
     ON DUPLICATE KEY
45     UPDATE
       Name = VALUES (Name),
47       Description = VALUES (Description),
       Distributed = VALUES (Distributed),
49       sourceAddr = VALUES (sourceAddr),
       insertTime = VALUES (insertTime);
51
53   </xsl:when>
     </xsl:choose>
55 </xsl:template>
57 </xsl:stylesheet>

```

Listing 1 AdminDomain.xslt

same identifier is already stored in the database, the information is updated (lines 44–50) instead of inserted.

The monitoring system of the Globus Toolkit 4 middleware, MDS4, can be queried using a WS-RF interface which provides output in XML format. As MDS4 is based on GLUE 1.1 and additional information providers like GeoMaint [44] provide site related information in GLUE 1.3, the XSLTs of the MDS4 gateway are implemented to parse, check and map the fields with respect to the different versions of GLUE.

The loader components of the implemented gateways store the transformed, homogeneous monitoring information in the D-MON integrated database.

6.3 VO-aware Monitoring

The VO-specific monitoring components extend the core D-MON system. This subsection covers its implementation details.

6.3.1 Serving VO-specific Views

In order to provide VO-based monitoring, information about the VOs has to be obtained from the VO resource management system. In D-Grid, the Grid Resource Registration Service (GRRS) holds the mappings between resources and VOs

and information about all available resources in the Grid. Whenever a gateway inserts new information into the D-MON database, a trigger is activated and generates information for the GLUE 2.0 AccessPolicy table from the GRRS tables. An extract of this trigger handling MDS4 specific mappings is shown in Listing 2.

In a second step VO-specific database views are generated. To keep this step as flexible as possible a dynamic view is set up, which can be parameterized. This avoids manual intervention whenever a new VO is created or an existing one is deleted. An SQL session parameter containing the VO has to be set when querying the database to retrieve the VO-specific data. Listing 3 shows how such views are realized for the GLUE 2.0 tables Endpoint, ComputingEndpoint and ComputingService. In line 4 the tables Endpoint and AccessPolicy are joined on the attribute serviceID. Afterwards the joined data is filtered to contain the data of the specified VO only. Therefore the session parameter is read by the database function *getVO()*. This function had to be implemented as MySQL does not offer it. The same procedure is realized in line 8 using the ComputingEndpoint table instead of Endpoint. The third view (lines 10 to 16) containing VO-aware information about ComputingServices is created by a join with the VO-aware view of the ComputingEndpoint instead of AccessPolicy because a direct join is impossible.

```

1 CREATE TRIGGER 'setAccessPolicy' AFTER INSERT ON 'ComputingEndpoint'
  FOR EACH ROW BEGIN
3   CASE NEW.informationProvider
5     WHEN "MDS4" THEN
      REPLACE INTO GLUE20.AccessPolicy (serviceID, localID, userDomainID,
        endpointID, informationProvider, sourceAddr, insertTime)(
7       SELECT NEW.cServiceID, b.driv_vo, b.driv_vo, NEW.endpointID, "TRIGGER", "
        127.0.0.1", CURRENT_TIMESTAMP
      FROM GRRS.resources a, GRRS.resource_vo b
9       WHERE a.resource_key=b.resource_key
        AND a.globus_resource LIKE CONCAT("%",NEW.cServiceID,"%")
11    );
13   WHEN "BDII" THEN
      REPLACE INTO GLUE20.AccessPolicy ...
15   WHEN "CIS" THEN
      REPLACE INTO GLUE20.AccessPolicy ...
19  END CASE;
    END;

```

Listing 2 setAccessPolicy.sql

Listing 3 voView.sql

```

CREATE VIEW 'GLUE20'. 'Endpoint_VO' AS (
2 SELECT a.*
FROM Endpoint a, AccessPolicy b
4 WHERE a.serviceID=b.serviceID and b.userdomainID=getVO());

6 CREATE VIEW 'GLUE20'. 'ComputingEndpoint_VO' AS (
SELECT a.* FROM ComputingEndpoint a, AccessPolicy b
8 WHERE a.cServiceID=b.serviceID and b.userdomainID=getVO());

10 CREATE VIEW 'GLUE20'. 'ComputingService_VO' AS (
SELECT * FROM ComputingService b WHERE EXISTS (
12 SELECT *
FROM ComputingEndpoint_VO a
14 WHERE a.cServiceID=b.ID
)
16 );
18 ...

```

6.3.2 Standardized OGSA-DAI Service Access Points for Secure VO-aware Queries

The standardized service access interface is implemented based on OGSA-DAI since SQL itself is not a Grid standard and is not able to handle Grid specific authentication and authorization protocols. OGSA-DAI is a standard to query SQL databases in a WS-RF-compatible way. The developed interface first sets the session parameter before VO-specific queries to the database are

performed. Future versions of OGSA-DAI will be capable of dealing with SAML. Then it will be possible to set the VO parameter in an automated manner as described in Section 5.3.2.

6.3.3 Standardized User Interface

Based on the standardized access client, a command line client has been developed. It copes with the requirement of being VO-aware by setting a SQL environment variable as a parameter to

Table 1 Test results with the OGSA-DAI client

| Entries for VO Kern D-Grid | | | | | |
|--|----------------------|-----------|-------------|-------------|---------------------|
| SQLQuery: SELECT name, type, totalJobs, runningJobs, waitingJobs, informationProvider FROM ComputingService_VO | | | | | |
| VO-Parameter: kg | | | | | |
| uk.org.ogsadai.resource.request.status.COMPLETED | | | | | |
| name | type | totalJobs | runningJobs | waitingJobs | informationProvider |
| aprilia.izbi.uni-leipzig.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| arminius-grid.uni-paderborn.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| dgiref-globus.fzk.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| dgrid-glitecl.rz.rwth-aachen.de | sge-dgiseq | 0 | 0 | 0 | BДИ |
| dgrid-globus.rz.RWTH-Aachen.DE | org.teragrid.ws-gram | 101 | 59 | 42 | MDS4 |
| dgrid-globus4.physik.uni-wuppertal.de | org.teragrid.ws-gram | 82 | 82 | 0 | MDS4 |
| globus-hn.mathematik.uni-marburg.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| gloria.d-grid.scai.fraunhofer.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| gramd1.d-grid.uni-hannover.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| gt4-fzk.gridka.de | org.teragrid.ws-gram | 8366 | 6145 | 2221 | MDS4 |
| gt4.dgrid.hlrs.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| hector.zih.tu-dresden.de | org.teragrid.ws-gram | 16 | 16 | 0 | MDS4 |
| iwrgrt4.fzk.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| juggle-glob.fz-juelich.de | org.teragrid.ws-gram | 118 | 109 | 9 | MDS4 |
| mardschana.zib.de | org.teragrid.ws-gram | 41 | 31 | 10 | MDS4 |
| medgrid-srv.gwdg.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| othello.zih.tu-dresden.de | org.teragrid.ws-gram | 16 | 16 | 0 | MDS4 |
| romeo.urz.tu-dresden.de | org.teragrid.ws-gram | 3 | 3 | 0 | MDS4 |
| sigrid.zimt.uni-siegen.de | org.teragrid.ws-gram | 71 | 32 | 39 | MDS4 |
| srvgrid01.offis.uni-oldenburg.de | org.teragrid.ws-gram | 186 | 153 | 33 | MDS4 |
| udo-gt01.grid.uni-dortmund.de | org.teragrid.ws-gram | 151 | 151 | 0 | MDS4 |
| udo-gt03.grid.tu-dortmund.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |

| Entries for VO Test | | | | | |
|--|----------------------|-----------|-------------|-------------|---------------------|
| SQLQuery: SELECT name, type, totalJobs, runningJobs, waitingJobs, informationProvider FROM ComputingService_VO | | | | | |
| VO-Parameter: te | | | | | |
| uk.org.ogsadai.resource.request.status.COMPLETED | | | | | |
| name | type | totalJobs | runningJobs | waitingJobs | informationProvider |
| arminius-grid.uni-paderborn.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| dgiref-globus.fzk.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |
| dgrid-glitecl.rz.rwth-aachen.de | sge-dgiseq | 0 | 0 | 0 | BДИ |
| dmon-unic.fz-juelich.de\$9115\$FZJ-JUGGLE-DMON | de.fzj.unicore.tsf | 0 | 0 | 0 | CIS |
| udo-gt01.grid.uni-dortmund.de | org.teragrid.ws-gram | 151 | 151 | 0 | MDS4 |
| udo-gt03.grid.tu-dortmund.de | org.teragrid.ws-gram | 0 | 0 | 0 | MDS4 |

process VO-specific information. A VO view is created by the database according to the parameter and the SQL query which has been sent to the OGSA-DAI Server. The outcome is a view, which only contains data sets that are assigned to the specific VO and to the local user or service. Table 1 shows the output of the OGSA-DAI client, monitoring computing services for two different VOs in three middlewares.

GridSphere was chosen as the portal framework as it complies with the needs of the Grid Community. GridSphere provides a sophisticated Portlet Service Mode that encapsulates the reusable portlet logic into services that may be shared between many portlets.

Various portlets are used to visualize the data in different views. This includes views like the classic table view, drill down or Google Maps. They are implemented dynamically using AJAX (Asynchronous JavaScript and XML) and Inter-Portlet Communication (IPC) to provide a more intuitive user experience whenever possible. The user's role is taken into account when building the view.

A SAML-VOMS authentication module, which automatically extracts the user information from the user's certificate and sends it to the SAML-VOMS server, will be added to the GridSphere Client in a next step. For this purpose the SAML-VOMS module of the Vine Toolkit [45] will be converted into an authentication module.

7 Summary and Outlook

In this paper we described a distributed architecture for an interoperable Grid monitoring system. It bases on unifying data from different middleware solutions into a standardized format and uses VOMS to support VO-centric, role-based views for data access from clients. All components selected for the system are standards either from the Open Grid Forum (OGF) or the Organization for the Advancement of Structured Information Standards (OASIS). The chosen approach enables the interoperation of existing Grid monitoring systems and proved to be viable.

The D-MON monitoring system has been deployed in the D-Grid environment. A testbed col-

lects the data from the three central information services of all Globus, Glite and UNICORE 6 sites (in total 115 sites with about 700 users). Upload of the monitoring data from the sites to the D-MON data base is done in five minute intervals. Currently we are in the process of further evaluating the performance of the different components of the D-MON monitoring system. As can be seen in Table 1 data from all three middleware installations is integrated into the GLUE 2.0-based database. The standardized data schema is a key to the unified provision of monitoring information from the different middleware solutions.

The data integration process revealed data inconsistencies which mostly resulted from data duplication through inconsistent naming of sites and resources. This is an administrative issue that must be solved by a naming and configuration authority, e.g. a Grid Operation Center. It can hardly be solved on a technical level. The data provided by the different information services is not sufficient to be used for accounting resource usage. Schedulers including middleware-specific schedulers can base their decisions on the data gathered in the the D-MON database. As a prove of concept the collected data is being fed back into the middleware-specific information system CIS for being used by local schedulers. Another experience gained so far is that with significant changes in the GLUE schema the database has to be set up from scratch and the old data is lost. Minor changes to the schema require only the adaptation of the ETL procedures for the different middlewares.

The system is still under development and the full integration of a SAML-VOMS server for authentication and authorization of data access is work in progress. We are currently extending the available database to include storage elements and evaluate federated setups, as well as different techniques regarding performance and stability.

Overall, the developed system is a good example for a non-intrusive, interoperable monitoring solution for heterogeneous Grid infrastructures. It enables monitoring systems of different middlewares to easily exchange data.

Acknowledgements The authors wish to thank the Grid communities and the members of the working groups related to OGF GLUE, OGSA-DAI, OMII, GLOBUS and UNICORE for helpful advice and support.

References

- Värttö, S. (ed.): DEISA Advancing Science in Europe. www.deisa.eu, <http://www.deisa.eu/press/Media> (2008)
- Appleton, O., Cook, A., Hämmerle, H.: EGEE in 2007. www.eu-egee.org, <http://press.eu-egee.org/fileadmin/documents/publications/EGEEin2007-final.pdf> (2008)
- TeraGrid: Open scientific discovery infrastructure. <http://www.teragrid.org/>. Accessed September 2008
- Neuroth, A., Kerzel, M., Gentzsch, W. (eds.): German Grid Initiative D-Grid. Universitätsverlag Göttingen (2007)
- Ellert, M., Gronager, M., Konstantinov, A., Konya, B., Lindemann, J., Livenson, I., Nielsen, J., Niinimäki, M., Smirnova, O., Waananen, A.: Advanced Resource Connector middleware for lightweight computational Grids. *Future Gener. Comput. Syst.* **23**(2), 219–240 (2007)
- D-MON Project: Horizontal integration of resource- and service monitoring in d-Grid. <http://www.d-grid.de/index.php?id=401&L=1>. Accessed September 2008
- Alef, M., Fieseler, T., Freitag, S., Garcia, A., Grimm, C., Gürich, W., Mehammed, H., Schley, L., Schneider, O., Volpato, G.: Integration of Multiple Middlewares on a Single Computing Resource. *Future Generation Computer Systems* (2008). Accessed May 2008
- Tierney, B., Aydt, R., Gunter, D., Smith, W., Swany, M., Taylor, V., Wolski, R.: A Grid Monitoring Architecture. Tech. Rep. GDF.7, Global Grid Forum (2002)
- Schopf, J., Raicu, I., Perlman, L., Miller, N., Kesselman, C., Foster, I., D'Arcy, M.: Monitoring and discovery in a web services framework: functionality and performance of Globus Toolkit MDS4. In: *Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing*. IEEE Computer Society, Los Alamitos (2006)
- Cooke, A., Gray, A.J.G., Ma, L., Nutt, W., Magowan, J., Oevers, M., Taylor, P., Byrom, R., Field, L., Hicks, S., Leake, J., Soni, M., Wilson, A., Cordenonsi, R., Cornwall, L., Djaoui, A., Fisher, S., Podhorszki, N., Coghlan, B., Kenny, S., OrsquoCallaghan, D.: R-GMA: An information integration system for Grid monitoring. In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. LNCS, vol. 2888, pp. 462–481. Springer, New York (2003)
- Foster, I., Maguire, T., Snelling, D.: OGSA WSRF Basic Profile 1.0. Tech. Rep. GDF.72, Global Grid Forum (2006)
- Foster, I.: Globus Toolkit Version 4: software for service-oriented systems. In: *IFIP International Conference on Network and Parallel Computing*. LNCS, no. 3779, pp. 2–13. Springer, New York (2006)
- gLite: Lightweight Middleware for Grid Computing. <http://glite.web.cern.ch/glite/>. Accessed September 2008
- Marzolla, M., Andreatto, P., Venturi, V., Ferraro, A., Memon, A.S., Memon, M.S., Twedell, B., Riedel, M., Mallmann, D., Streit, A., van de Berghe, S., Li, V., Snelling, D., Stamou, K., Shah, Z.A., Hedman, F.: Open standards-based interoperability of job submission and management interfaces across the Grid middleware platforms gLite and UNICORE. In: *E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, pp. 592–601. IEEE Computer Society, Washington, DC (2007). doi:10.1109/E-SCIENCE.2007.54
- Hegering, H.G., Abeck, S., Neumair, B.: *Integrated Management of Networked Systems: Concepts, Architectures, and Their Operational Application*. Morgan Kaufmann, San Francisco (1999)
- Adreozzi, S., Burke, S., Ehm, F., Field, L., Galang, G., Konya, B., Litmaath, M., Millar, P., Navarro, J. (ed.): *GLUE 2.0 Specification V2.0*. Tech. rep., Global Grid Forum (2008)
- EDG: European Data Grid Project. <http://eu-datagrid.web.cern.ch/>. Accessed October 2008
- DataTAG: Research & Technological Development for a Data TransAtlantic Grid. <http://datatag.web.cern.ch/>. Accessed October 2008
- iVDGL: International Virtual Data Grid Laboratory. <http://igoc.ivdgl.indiana.edu/>. Accessed October 2008
- LCG: Worldwide LHC Computing Grid. <http://lcg.web.cern.ch/LCG/>. Accessed October 2008
- OSG: Open Science Grid. <http://www.opensciencegrid.org/>. Accessed October 2008
- OGF: Open Grid Forum. <http://www.ogf.org>. Accessed October 2008
- Mach, R., Lepro-Metz, R., Jackson, S.: Usage Record Format Recommendation. Tech. rep., Open Grid Forum (2006)
- Wolf, A.: Spezifikation der D-Grid-Ressourcenbeschreibungssprache DGRDL. Tech. rep., Fraunhofer FIRST (2007)
- Chem, X., Khan, A., Ainsworth, J., Newhouse, S., MacLaren, J.: WSI Resource Usage Service (RUS) Core Specification. Tech. rep., Open Grid Forum (2007)
- Fielding, R.: Architectural styles and the design of network-based software architectures. Ph.D. Thesis, University of California, IRVINE (2000)
- Baur, T., Saad, S.: Customer service management for grid monitoring and accounting data. In: *Proceedings of the 18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2007)*. Springer, New York (2007)
- Baur, T.: Functional analysis and architecture for interoperable and DVO-specific Grid monitoring services. In: *Proceedings of the Fourth IEEE International Conference on eScience (eScience 2008)*. IEEE Computer Society, Los Alamitos (2008)
- UNICORE: Uniform Interface to Computer Resources. <http://www.unicore.eu/>. Accessed September 2008
- Fuhrmann, P.: dCache, the Overview. <http://www.dcache.org/manuals/dcache-whitepaper-light.pdf>. Accessed September 2008. White Paper
- Antonioletti, M., Atkinson, M.P., Baxter, R., Borley, A., Hong, N.C., Collins, B., Hardman, N., Hume, A., Knox, A., Jackson, M., Krause, A., Laws, S., Magowan, J., Paton, N., Pearson, D., Sugden, T., Watson, P.,

- Westhead, M.: The Design and Implementation of Grid Database Services in OGSA-DAI. *Concurrency and Computation: Practice and Experience* **17**(2–4), 357–376 (2005)
32. Schopf, J.M., Pearlman, L., Miller, N., Kesselman, C., Foster, I., D'Arcy, M., Chervenak, A.: Monitoring the grid with the Globus Toolkit MDS4. *Journal of Physics Conference Series* **46**, 521–525 (2006). doi:10.1088/1742-6596/46/1/072
33. Duarte, A., Nyczyk, P., Retico, A., Vicinanza, D.: Monitoring the EGEE/WLCG grid services. *Journal of Physics: Conference Series* **119**(5), 052,014 (10pp) (2008). <http://stacks.iop.org/1742-6596/119/052014>
34. Memon, A.S., Memon, M.S., Wieder, P., Schuller, B.: CIS: An Information Service based on the Common Information Model. In: *Proceedings of 3rd IEEE International Conference on e-Science and Grid Computing*, pp. 465–472. IEEE Computer Society, Los Alamitos (2007)
35. Felde, V.D.G.O., Baur, T., Garschhammer, M., Reiser, H.: Anforderungen an das monitoring, ergebnisse aus der erhebung bei den communities und ressourcenanbietern im D-Grid. In: Rückemann, C.-P. (ed.) *Ergebnisse der Studie und Anforderungsanalyse in den Fachgebieten Monitoring, Accounting, Billing bei den Communities und Ressourcenanbietern im D-Grid (D-Grid Report)*. http://www.d-grid.de/fileadmin/dgi_document/FG2/koordination_mab/mab_studie_ergebnisse.pdf (2006). Accessed October 2008
36. Baur, T., Felde, V.D.G.O., Reiser, H.: Konzepte zum Monitoring im D-Grid (D-Grid Report). http://www.d-grid.de/uploads/media/mab_monitoring_konzepte.pdf (2007). Accessed October 2008
37. Kunze, M., Pogoshyan, G.: Monitoring for multi-middleware grid. In: *IEEE International Symposium on Parallel and Distributed Processing*, 2008. IPDPS 2008. IEEE Computer Society, Los Alamitos (2008)
38. Simmhan, Y.L., Plale, B., Gannon, D.: *A Survey of Data Provenance Techniques*. Tech. Rep. TR-618, Computer Science Department, Indiana University, Bloomington (2005)
39. Berry, D., Luniewski, A., Antonioletti, M.: *OGSA Data Architecture*. Tech. Rep. GDF.121, Global Grid Forum (2007)
40. Baur, T., Kalman, T., Lindinger, T., Milbert, A., Poghosyan, G., Romberg, M.: *Middleware-übergreifendes Monitoring: Evaluierung und Auswahl von Komponenten (D-Grid Report)*. http://www.d-grid.de/fileadmin/user_upload/documents/D-MON/Komponenten-AP2.1-final.pdf. Accessed September 2008
41. Chaudhuri, S., Dayal, U.: Data warehousing and OLAP for decision support. In: *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pp. 507–508. ACM, New York (1997). doi:10.1145/253260.253373
42. Venturi, V., Riedel, M., Memon, A.S., Memon, M.S., Stagni, F., Schuller, B., Mallmann, D., Tweddell, B., Gianoli, A., Ciaschini, V., van de Berghe, S., Snelling, D., Streit, A.: Using SAML-based VOMS for Authorization within Web Services-based UNICORE Grids. In: Bougé, L., Forsell, M., Träff, J.L., Streit, A., Ziegler, W., Alexander, M., Childs, S. (eds.) *Euro-Par 2007 Workshops: Parallel Processing*. LNCS, no. 4854, pp. 112–120. Springer, New York (2008)
43. Adreozzi, S., Burke, S., Ehm, F., Field, L., Galang, G., Konya, B., Litmaath, M., Millar, P., Navarro, J. (ed.): *GLUE v. 2.0—Reference to Concrete Data Models*. Tech. Rep., Global Grid Forum (2008)
44. GeoMaint Project: Informationprovider (Sensor) for Geolocation and Maintenance Data. <http://geomaint.sourceforge.net/>. Accessed September 2008
45. The Vine Toolkit: <http://www.gridisphere.org/gridisphere/guest/vine/r/>. Accessed September 2008