

Migration to the GLUE 2.0 information schema in the LCG/EGEE/EGI production Grid

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2011 J. Phys.: Conf. Ser. 331 062004

(<http://iopscience.iop.org/1742-6596/331/6/062004>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 95.172.225.198

The article was downloaded on 23/01/2012 at 08:24

Please note that [terms and conditions apply](#).

Migration to the GLUE 2.0 information schema in the LCG/EGEE/EGI production Grid

Stephen Burke^{1,3}, Laurence Field² and David Horat²

¹ Rutherford Appleton Laboratory, Didcot, OX11 0QX, UK

² CERN, CH-1211 Genève 23, Switzerland

E-mail: stephen.burke@stfc.ac.uk

Abstract. The GLUE information schema has been in use in the LCG/EGEE production Grid since the first version was released in 2002. In 2008 a major redesign of GLUE, version 2.0, was defined in the context of the Open Grid Forum. The implementation of the publication and use of the new schema is a complex process which needs to be carefully managed to avoid any disruption to the production service, especially in the light of the end of the EGEE project and the transition to the new middleware and operational structures in EGI. In this paper we discuss the LDAP rendering of the schema, the upgrading of the Grid information system to allow both schemas to be used in parallel, the design and rollout of information providers, and the plans for migrating client software which uses the schema information. In particular we consider the implications for the specific requirements of the LCG project, especially regarding storage systems, accounting and monitoring.

1. Introduction

Version 1 of the GLUE information schema has been in use in the LCG/EGEE production Grid since 2002, and hence a large amount of experience of its strengths and weaknesses has been obtained, as reported at CHEP 07 [1]. The process to define a major new version 2.0 of the schema [2] during 2008/9 in the context of an Open Grid Forum (OGF) working group [3] was reported at CHEP 09 [4]. The new schema is not backward-compatible with the old one, and hence its introduction in a production Grid must be carefully managed to avoid disruption to existing clients. This paper reviews the structure of the new schema, discusses the definition of the LDAP rendering, and describes the process by which the new schema is being introduced into the production Grid, together with the progress achieved at the time of writing.

Note that in this paper LCG/EGEE/EGI refers specifically to the Grid operated jointly by those projects; the worldwide LCG collaboration [5] also uses other Grids, notably OSG [6] and NDGF [7]. EGEE [8] was succeeded by EGI [9] in May 2010.

2. The history of the GLUE schema

A Grid consists of a highly heterogeneous collection of services and resources controlled by a large number of service providers. There is therefore a need to collect information relating to their existence, properties and current state, and enable it to be queried by clients at any location. The format of this information is specified by an *information schema*, which defines a set of classes with

³ To whom any correspondence should be addressed.

attributes and relations. The transport of the information is mediated by an *information system*, which for the LCG/EGEE/EGI Grid is currently provided by the BDII [10], which is based around a set of LDAP servers. These concepts are logically separate; the same schema may have representations using many different technologies, and the same information system may transport information defined in multiple schemas.

2.1. GLUE 1.x

The GLUE (Grid Laboratory for a Uniform Environment) project was a collaboration between the EU-funded European DataGrid (EDG) and DataTAG projects and the US iVDGL, together with Globus [11]. The initial specification for version 1.0 of the GLUE information schema was published in September 2002, and some minor improvements resulted in version 1.1 in April 2003. Experience with the 1.1 schema over the following year or so led to a number of changes being proposed by LCG, and version 1.2 of the schema was defined in February 2005. A further evolution to version 1.3 [12] was agreed in October 2006, largely to improve the description of storage systems making use of the Storage Resource Manager (SRM) [13] technology.

A major constraint on the evolution of the schema was that changes were always required to be backward-compatible with the previous versions. This facilitated the transitions from one version to the next, but in some areas it imposed a major constraint on the degree to which problems could be solved.

2.2. GLUE 2.0

The process to define a major new version of the schema in the context of an OGF working group began in January 2007, and a draft document was released for public comment in June 2008. The resulting comments were addressed and a final version was produced in January 2009, which was approved at the OGF meeting in the following March.

The overall structure of the new schema is as follows. The **UserDomain** and **AdminDomain** classes contain information about Virtual Organisations (VOs) and sites/resource providers respectively; in either case the objects of these types can form a hierarchy that describes their composition. The rest of the schema represents the general concept of a Grid **Service** as a coherent aggregation of network **Endpoints**, hardware **Resources** and software **Managers**. **Endpoints** have an **AccessPolicy** which specifies who is authorized to use the service, and authorized clients (users) are allocated a **Share** of the underlying resources as defined by a **MappingPolicy**. Finally, the interaction of a user with the service may be represented as an **Activity**. All classes are extensible, with a multi-valued string attribute “**OtherInfo**” and/or key/value pairs (**Extension** classes). All objects have a globally unique **ID** attribute, hence arbitrary references and queries are possible.

This structure is intended to be applicable to any Grid service, although not all classes may be useful in a specific case, and conversely it may be necessary to define additional classes for more specialised services. The general structure can also be extended by deriving new classes from the existing ones; the current schema defines such specialisations for computing and storage services.

3. The LDAP rendering

The abstract schema may be represented in a number of different implementation formats, and these are also defined by the GLUE working group. The current intention is to define representations at least for LDAP, XML and SQL, of which LDAP is the principal technology used in the LCG/EGEE/EGI Grid and the first one to be completed.

To render the schema in a given technology it is necessary to define a general way to map the abstract schema elements into the concepts provided by that technology. This can potentially be difficult if there is a substantial mismatch between the underlying conceptual models. However, while defining the GLUE schema it was known that LDAP would be one of the most important implementation technologies, and we have a great deal of experience with the LDAP rendering for GLUE 1.x, so in practice the mapping can be done in a rather natural way. The rendering was defined

rapidly, at least by the standards of the overall schema process, in half a dozen phone meetings during May and June 2009, together with email discussion. In most cases we broadly followed the GLUE 1.x practice, but there are a few significant changes:

- GLUE 2 defines string attributes to be case-sensitive, and the LDAP rendering enforces that, whereas the GLUE 1 rendering uses case-insensitive strings.
- Attributes marked as mandatory in the schema are mandatory in LDAP; in the GLUE 1 rendering nearly all attributes are optional even if they are in practice essential.
- In GLUE 1 the string type used (IA5String) is restricted to 7-bit ASCII characters, which has sometimes caused problems because strings containing non-ASCII characters, e.g. with non-English locales, cause the entire object to be rejected as invalid. For GLUE 2 we have taken the opportunity to switch to the DirectoryString type which allows the extended UTF-8 character set. (The GLUE 2 schema document itself is silent on this question, which is perhaps an omission.)
- The naming and usage of ForeignKey attributes used to represent relations is somewhat different, as described below.

The formal definition of the LDAP rendering is described in an OGF draft document [14]; the main points are briefly summarised here.

3.1. Object classes

Object classes are the basic structural unit of an LDAP schema. We take the natural mapping that each schema entity corresponds to one LDAP object class. Inheritance in the schema is represented by composition, i.e. in general an LDAP object instance includes attributes from several object classes. The object class names follow the schema entity names; however, LDAP has no explicit namespacing, so to avoid any possibility of name clashes we prefix all names with the string “GLUE2”. In particular this avoids clashes with the GLUE 1.x schema where the prefix is “Glue”. So for example, the abstract **ComputingShare** entity becomes an LDAP object class called **GLUE2ComputingShare**.

3.2. Attributes

In general, LDAP attribute naming extends the naming rule for object classes, i.e. the name is “GLUE2” + *<schema entity name>* + *<schema attribute name>*, for example **GLUE2ComputingShareMaxCPUTime**. However, we made an exception for the **ID** attribute, which as an attribute of every object is defined in the **Entity** class. As described below the object DN is constructed from the **IDs**, and if we had followed the general rule we would have produced DNs of the form **GLUE2EntityID=x**, **GLUE2EntityID=y**, **GLUE2EntityID=z** which seems unduly opaque. We therefore name the attribute from the first-level object derived from **Entity**, i.e. a **ComputingShare** object has an ID called **GLUE2ShareID** (rather than **GLUE2EntityID** or **GLUE2ComputingShareID**).

The LDAP attribute multiplicity directly follows the multiplicity defined in the schema, i.e. attributes are single- or multi-valued and optional or mandatory as appropriate.

One limitation of LDAP is that it has rather limited support for data types, so all non-numeric attributes are simply typed as LDAP strings, implying that any validation must be performed with an external tool.

3.3. Foreign keys

These represent relations between objects, something which is not explicitly available in LDAP, so we need to define attributes with a specific convention for naming and semantics. For GLUE 1 we have a generic attribute called **GlueForeignKey** which can be used in any object. This contains a string of the form “**GlueSiteUniqueID=<site name>**” which can be directly used as a filter in an LDAP query. However, this has the disadvantage that relations which are not defined in the schema can be

published by accident, and also makes querying and parsing somewhat more complex because all relations have the same attribute name. For GLUE 2 we therefore decided to have an explicit named attribute for every relation defined in the schema. These have long but unambiguous names of the form **GLUE2ComputingShareComputingServiceForeignKey**, representing a relation from **ComputingShare** to **ComputingService**, whose value is the **ID** of the related object. These attributes are inherited by derived objects in the same way as other attributes.

All relations defined in the schema have a corresponding **ForeignKey** attribute, even if they are also implied by the DN (see below). Relations have two ends, but the nature of LDAP queries is such that a **ForeignKey** attribute is only needed in one of the related objects. The object which carries the **ForeignKey** was decided case by case, but the general principle is that they point logically upwards, i.e. objects point to their parent to the extent that such a concept makes sense.

3.4. DN construction

LDAP objects are organised in a hierarchical tree, and each object has a Distinguished Name (DN) representing its location in the tree. The DN is an ordered list of attribute name/value pairs which must result in a unique DN for every object. For GLUE 2 the natural attribute to use is the **ID**, which is defined to be globally unique across all published objects. The GLUE schema relations are more complex than a tree can represent, and conversely the concept of a DN has no relevance to the abstract schema, so the organisation of the tree is largely driven by practical considerations. In particular, any navigation between objects should use the **ForeignKey** attributes and not rely on the DN. The specification of the LDAP rendering gives a suggested organisation for the tree based on a natural hierarchy insofar as it exists, but implementations are free to modify this subject to a few restrictions, e.g. that all objects relating to a particular **Service** should be grouped under that **Service**.

We also define a dummy grouping object whose only function is to allow an attribute **GLUE2GroupID** to be inserted anywhere in the DN; this has no semantics but may make the tree easier to manage, e.g. when viewing it in an LDAP browser.

An example of a possible DN for the **MappingPolicy** attached to a given **ComputingShare** might therefore be: **GLUE2PolicyID=x, GLUE2ShareID=y, GLUE2GroupID=V, GLUE2ServiceID=z, GLUE2DomainID=S, GLUE2DomainID=G, GLUE2GroupID=grid, o=glue**, where **x**, **y** and **z** are arbitrary unique identifiers, **V** is a VO name, **S** is a site name and **G** is a Grid name.

4. BDII integration and deployment

Having defined the LDAP schema it must be integrated into the BDII. This needs to be done in such a way that the chance of any disturbance to the production system is minimised. The method chosen is to combine the GLUE 1.3 and 2.0 schemas in a single LDAP server listening on a single port as before, but with separate root DNs for each schema version. The existing GLUE 1.3 deployment uses **o=grid** as the root, so for 2.0 we chose **o=glue**. This means that there is no chance of crosstalk between queries to the two schemas, so the only possible interference is due to the increase in the total data volume, which should be manageable. This also means that it is not possible to do single queries against both schemas simultaneously, but given the very different structure this would be unlikely to be useful in any case.

The current deployment of the BDII infrastructure has three levels: resource BDIIs contain information related to an individual service, site BDIIs aggregate all the resource information from services at a single site, and top-level BDIIs aggregate information from all sites in the Grid. Each of these presents the information under a specific base DN. For GLUE 2 we follow the same scheme, mutatis mutandis:

GLUE2GroupID=resource, o=glue (Resource BDII)

GLUE2DomainID=<site name>, o=glue (Site BDII)

GLUE2GroupID=grid, o=glue (Top-level BDII)

The implementation of the new LDAP schema and its deployment with the BDII in the production Grid followed rapidly after the LDAP rendering was defined, and was in production by September 2009. Upgrading the site and top-level BDIIs to aggregate the lower-level information took somewhat longer, but the full infrastructure was in place by October 2010. This means that there is now a complete GLUE 2 information system in the production Grid, although sites can be fairly slow to upgrade – at the end of 2010 the coverage was 58 sites out of 369.

5. Information providers

Without information providers to publish information in the GLUE 2 format the information system would be empty, and it is therefore necessary to upgrade all providers to publish in both formats. The easiest one to upgrade was the generic provider for the rather simple **GlueService** object in GLUE 1. This was initially modified to publish identical information in the new format, and then improved to allow publication of the more significant new attributes in GLUE 2; the first version was available in October 2009. During 2010 this was progressively rolled out as new versions of each service which uses the provider were released. By the end of 2010 six services were publishing in production, and others are in the pipeline.

Some services, notably the LCG File Catalog and File Transfer Service, have dedicated providers which need more work. Upgrades are now in progress and are expected to be in production by mid-2011.

Publication of information for computing and storage services is much more complex. In the computing case there is only one target service (CREAM), but it has interfaces to several batch systems, each of which needs to publish its own information. Implementation is in progress, and an initial version of the **Service** and **Endpoint** publication via the generic service provider is already in production. This is likely to be expanded by mid-2011 but probably with largely static information at first; full dynamic publication comparable with the GLUE 1 information may be available by the end of the year.

In the storage case we have multiple SRM implementations, each of which will need its own publisher. Work is now under way to develop prototype publishers for several of these implementations which will simply publish existing GLUE 1 information in the GLUE 2 format, and in most cases these are again expected to be available in some form by mid-2011. However, GLUE 2 was designed to allow a significantly better representation of the features available with version 2.2 of the SRM protocol than was possible in GLUE 1. The GLUE 2 publication will therefore need to evolve in the light of experience, particularly as GLUE 2-aware data management client tools are developed.

6. Clients

Once information is published according to the new schema, client tools can be migrated to use it. This will need to be done in a backward-compatible way to fall back to the existing information for sites/services which have not upgraded to the new publishers. At present, other than direct use of LDAP tools the only GLUE 2-aware client is the OGF service discovery API [15]. Major middleware clients are the Job Description Language of the Workload Management System (computing services) and the various data management tools (storage services). Experience suggests that migration will be a slow process, with a possible target date being the end of 2012 for the major clients to be fully GLUE 2-aware.

Within the production Grid the information system is also used as a source of information for monitoring the overall state of the Grid, and for high-level resource accounting, e.g. to determine the total installed computing and storage capacity. This is primarily via the GStat tool [16], so this will also need to be upgraded, both to gather and display information and to monitor the deployment and accuracy of the GLUE 2 publication itself. There may also be use cases for the publication of Grid management information which is not possible to publish with GLUE 1, since the addition of new information without disrupting the existing uses is much easier with the new schema.

7. Conclusions

The GLUE schema has developed over nine years of use by the EDG, LCG and EGEE projects. GLUE 2.0 is a major new schema version that incorporates our experience, and also collects input from many other Grids. It should cover all current use cases for the LCG/EGEE/EGI Grid, as well as allowing things we are unable to do using the 1.3 schema. The new schema provides a uniform structure to describe any service, and can be extended in a standardised way. It will also be much more flexible for new use cases that have not been anticipated but which experience suggests are likely to occur.

The implementation and deployment of the new schema started in 2009, and by the end of 2010 a GLUE 2 information system was in place with a number of services being published. This process will continue during 2011, and by the end of the year we hope to have all services publishing, at least in a preliminary form.

The transition to a new project structure is providing an additional impetus to the migration. The middleware is now supplied by the EMI project [17], which incorporates the ARC, Unicore and dCache distributions as well as gLite, and which is committed to adopting GLUE 2.

At some point it may be possible to remove the GLUE 1.3 information completely. However, this could only be done once we are sure that nothing relies on it, and this is likely to take several years. It can be seen that the full migration process will be rather slow, but for the reasons described above we believe that the potential gains are sufficient to justify making the transition.

References

- [1] Burke S, Andreozzi S and Field L 2008 Experiences with the GLUE Information Schema in the LCG/EGEE Production Grid *J. Phys.: Conf. Ser.* **119** 062019
- [2] Andreozzi S, Burke S, Field L, Galang G, Konya B, Litmaath M, Millar P and Navarro JP 2009 GLUE Specification v. 2.0 *OGF Proposed Recommendation GFD 147*
<http://www.ogf.org/documents/GFD.147.pdf>
- [3] GLUE OGF working group <http://forge.ogf.org/sf/projects/glue-wg>
- [4] Burke S, Andreozzi S, Donno F, Ehm F, Field L, Litmaath M and Millar P 2010 The impact and adoption of GLUE 2.0 in the LCG/EGEE Production Grid *J. Phys.: Conf. Ser.* **219** 062005
- [5] LCG (LHC Computing Grid) <http://www.cern.ch/lcg>
- [6] OSG (Open Science Grid) <http://www.opensciencegrid.org/>
- [7] NDGF (Nordic DataGrid Facility) <http://www.ndgf.org/>
- [8] EGEE (Enabling Grids for E-science) <http://www.eu-egee.org/>
- [9] EGI (European Grid Infrastructure) <http://www.egi.eu/>
- [10] BDII <https://twiki.cern.ch/twiki/bin/view/EGEE/BDII>
- [11] The Globus Alliance <http://www.globus.org/>
- [12] Andreozzi S, Burke S, Donno F, Field L, Fisher S, Jensen J, Konya B, Litmaath M, Mambelli M, Schopf J, Viljoen M, Wilson A and Zappi R 2007 GLUE Schema Specification - Version 1.3 - draft 3
http://forge.cnaf.infn.it/plugins/scmsvn/viewcvs.php/*checkout*/v_1_3/spec/pdf/GLUESchema.pdf?rev=49&root=glueschema
- [13] Storage Resource Management Working Group <http://sdm.lbl.gov/srm-wg/>
- [14] Andreozzi S, Burke S, Ehm F, Field L, Galang G, Horat D, Konya B, Litmaath M, Millar P and Navarro JP 2010 GLUE v. 2.0 – Reference Implementation of an LDAP Schema *OGF Draft Recommendation*
<http://forge.ogf.org/sf/go/doc15518?nav=1>
- [15] Fisher F, Wilson A and Paventhan A 2009 SAGA API Extension: Service Discovery API *OGF Proposed Recommendation GFD 144*
<http://www.ogf.org/documents/GFD.144.pdf>
- [16] GStat <http://gstat-prod.cern.ch/gstat/summary/>
- [17] EMI (European Middleware Initiative) <http://www.eu-emi.eu/>