# The impact and adoption of GLUE 2.0 in the LCG/EGEE production Grid

**Stephen Burke[1,5], Sergio Andreozzi[2], Flavia Donno[3], Felix Ehm[3], Laurence Field[3], Maarten Litmaath[3] and Paul Millar[4]**

[1] Rutherford Appleton Laboratory, Didcot, OX11 0QX, UK
[2] INFN-CNAF, Viale Berti Pichat 6/2, 40126 Bologna, Italy
[3] CERN, CH-1211 Genve 23, Switzerland
[4] DESY, Notkestraße 85, D-22603, Hamburg, Germany

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 defined in 2002. In 2007 a major redesign of GLUE, version 2.0, was started in the context of the Open Grid Forum following the creation of the GLUE Working Group. This process has taken input from a number of Grid projects, but as a major user of the version 1 schema LCG/EGEE has had a strong interest that the new schema should support its needs. In this paper we discuss the structure of the new schema in the light of the LCG/EGEE requirements and explain how they are met, and where improvements have been achieved compared with the version 1 schema. In particular we consider some difficulties encountered in recent extensions of the use of the version 1 schema to aid resource accounting in LCG, to enable the use of the SRM version 2 storage protocol by the LHC experiments, and to publish information about a wider range of services to improve service discovery. We describe how these can be better met by the new schema, and we also discuss the way in which the transition to the new schema is being managed.

## 1. Introduction

Version 1 of the GLUE information schema has been in use in the LCG/EGEE production Grid for many years, and hence a large amount of experience of its strengths and weaknesses has been obtained, as reported at CHEP 07 [1]. The schema had been extended several times over the years, but always incrementally, with the constraint that any changes were backward-compatible. In 2007 the time seemed right to start work on a completely new version of the schema, which would have no compatibility requirements and hence could be completely restructured, allowing many outstanding problems to be tackled that were difficult or impossible to solve within the existing schema structure. The opportunity was also taken to define the new schema version in the context of an Open Grid Forum (OGF) working group [2], leading to buy-in from a much wider range of Grid projects which enables GLUE to become a true Grid standard. This process has taken two years to come to fruition, but version 2.0 of GLUE has now been finalized [3], and the start of its deployment in the LCG/EGEE infrastructure is imminent.

[5] To whom any correspondence should be addressed.

 

This paper reviews the need for an information schema and the historical development of the GLUE schema, and discusses some aspects of the experience gained within the LCG/EGEE Grid, and in particular those areas where the structure of version 1 of the schema created problems which were difficult to solve with minor revisions. We then describe the process by which the new schema was defined, and explain the main features of version 2.0 of the schema. Finally, we consider the process by which the new schema will be introduced in the LCG/EGEE Grid, and the principle advantages it will bring relative to the current schema.

Note that in this paper LCG/EGEE refers specifically to the Grid operated jointly by those two projects [4],[5]; the worldwide LCG collaboration also uses other Grids, notably OSG [6] and NDGF [7].

## 2. The need for an information 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 transport of this information is mediated by an *information system*, which for the LCG/EGEE Grid is currently provided by the BDII [8], which is a system of LDAP servers. The format of the information is specified by an *information schema*, which defines a set of classes with attributes and relations. These concepts are logically separate; the same schema may have representations using many different technologies.

In the LCG/EGEE Grid, the information system is used to satisfy several general categories of use case. It must allow the discovery of all Grid services and resources, and publish attributes that allow clients to select those that best meet their requirements. It enables a variety of monitoring tools to present an overview of the state of the Grid. Finally, it is currently used for coarse-grained resource accounting, although accounting is not in general an intended use for the system.

Grid resources are highly varied, and the information schema therefore needs to be flexible enough to represent that variety. However, to be useful the schema also needs to represent the information in a standardised, uniform way. These requirements represent a fundamental tension, which inevitably means that on the one hand many details cannot be represented in the schema, and on the other hand that some aspects will appear complex to users. The balance struck between these requires judgements about the practicalities of use in real Grids, and hence has evolved with experience.

Different Grids may have differing requirements for their information schemas. However, inevitably they also have many things in common, and in addition it is desirable to enable interoperability between Grids, especially for the LCG/EGEE Grid which has relationships with many other Grid projects. Therefore, it is useful to standardise the schema where possible. This standardisation has progressively increased over the years as Grid projects have gained more understanding and experience.

## 3. The history of the GLUE schema

The GLUE (Grid Laboratory for a Uniform Environment) project was a collaboration between the EU-funded European DataGrid (EDG) [9] and DataTAG [10] projects and the US iVDGL [11], together with Globus [12]. The GLUE information schema was developed from the custom schema used by the EDG project from 2001, and the initial specification for version 1.0 of the schema was published in September 2002. Some minor improvements resulted in version 1.1 [13] in April 2003, and this version was deployed in production by EDG and subsequently LCG/EGEE. Experience with the 1.1 schema over the following year or so led to a number of changes being proposed by LCG in November 2004, and version 1.2 of the schema [14] was agreed in February 2005 and deployed by LCG/EGEE during 2006. A further evolution to version 1.3 [15] was agreed in October 2006, largely to improve the description of storage systems making use of the Storage Resource Manager (SRM) [16] technology. This was progressively deployed by LCG/EGEE during 2007/8.

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. Also, although some attributes were deprecated it has not so far proved possible in practice to remove most of them, resulting in an accumulation of obsolete or redundant attributes.

### 3.1. Problems with version 1.x of the GLUE schema

A previous CHEP paper [1] discussed the LCG/EGEE experience with the 1.x GLUE schema versions in some detail, but some of the key points are recapitulated here:

- The structure of the computing part of the schema does not have a clear separation between the concepts of the batch system itself, and queues and fairshare targets defined within it. An attempt was made to improve this situation with the 1.2 schema revision, but the requirement of backward compatibility meant that this could only be addressed in a limited way.
- The definition of the computing hardware assumed that it could be described in terms of several sets of homogeneous nodes, but in practice the Workload Management System used in LCG/EGEE has not been able to use this information as intended.
- The initial description of storage systems was defined before we had any experience with the SRM protocol, and hence was not well-adapted for it. Although subsequent schema revisions have improved the situation, the requirement of backward compatibility places substantial constraints on what could be represented, particularly in the context of SRM space tokens, which are now heavily used in the production Grid.
- The first version of the schema had representations only of computing and storage services. Version 1.2 introduced a representation of generic Grid services, but this does not directly connect with the pre-existing computing and storage representations, and is rather limited in terms of what information can be published.
- There are also many cases where the mapping between GLUE attributes and real systems could be ambiguous or where embedded assumptions are too restrictive, particularly where technology has advanced; e.g., with the introduction of multi-core CPUs and virtual machines. Many things are only defined by the current practice in LCG/EGEE, rather than being specified explicitly.

## 4. GLUE 2.0

By the time of the meeting in October 2006 which agreed version 1.3 of the schema it had become apparent that the time was right to consider a major, non-backward-compatible revision. This was both because a number of outstanding problems could not be solved within the existing schema structure, and because Grids had matured sufficiently that we could design a schema whose structure was likely to suffice for some time to come. It was also desirable to standardise the schema on a wider scale than the existing users of the version 1.x schemas, and it was therefore decided to create a GLUE working group within the OGF.

### 4.1. The GLUE 2.0 process

The first meeting of the GLUE working group was in January 2007. The working group interacted with some 14 Grid projects, although LCG and EGEE were the major contributors. This initiated a process largely conducted by email and phone meetings, which led to a draft document 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 25 meeting in the following March.

It may be interesting to give a few statistics to illustrate what was involved:

- There were 60 phone conferences lasting 1.5-2 hours each, so around 5 days in total. There were also around 1000 emails in the GLUE mailing list.

- There were typically 5 people in each meeting, so the phone meetings alone consumed about 4 months FTE.
- There were also 18 sessions over 7 OGF meetings, with a wider participation from the general OGF community.
- There was in addition a substantial effort from the editor of the draft document, of which there were 50 versions. The document is 76 pages long, with 27609 words.
- The public comment period generated about 40 new issues from 10 people, to add to the 60 which had been carried over from previous discussion.
- The schema defines 246 attributes in 35 classes, not counting relations or inherited attributes.

Moving GLUE into the OGF has meant that the process has been somewhat slower than in the past as it has involved discussion with a wider range of people, who sometimes have divergent views, and the schema itself is somewhat more complex as it contains features to support the use cases of particular Grids. However, these are also strongly positive points: scrutiny from a wider range of viewpoints has helped to clarify many issues, and the schema is likely to be used and supported by a much wider range of projects. GLUE also interacts with other OGF standardization activities, and the profile of GLUE has been raised even within EGEE itself.

### 4.2. The schema structure

The overall structure of the new schema is illustrated in figure 1. 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 **Endpoint**s, hardware **Resource**s and software **Manager**s. **Endpoint**s have **AccessPolicy**s which specify 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**.

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 specialized services. The general structure can be specialized by deriving new classes, and the current schema defines such specializations for Computing Services and Storage Services.

The Computing Service (figure 2) introduces one additional class to represent the Application Environment; i.e., the software environment in which the jobs may run. The interpretation of the standard classes is that the **ComputingManager** represents the batch system, the **ExecutionEnvironment** (**Resource** specialization) represents the hardware and operating system characteristics of a type of Worker Node, and the **ComputingActivity** represents a batch job. The **ComputingShare** represents a queue and/or a fairshare allocation within the batch system.

The Storage Service (figure 3) introduces additional **Capacity** classes to carry information about the size and usage of storage space, both at the level of the entire storage system (**StorageCapacity**) and at a finer granularity (**ShareCapacity**). These have some scope to provide information for an accounting service, but they cannot be used to account for usage by individual users or to provide information at the level of individual files or transactions (writing or deleting data). The schema also defines an **AccessProtocol** class to provide information about protocols used to access (i.e. read or write) data; these cannot in general be described by the **ServiceEndpoint** class as the endpoints are often generated dynamically for each transaction. The specialisations of the standard classes represent the data storage hardware (**DataStore**, derived from **Resource**), the storage software (**StorageManager**) and the space allocations to **UserDomain**s (**StorageShare**).

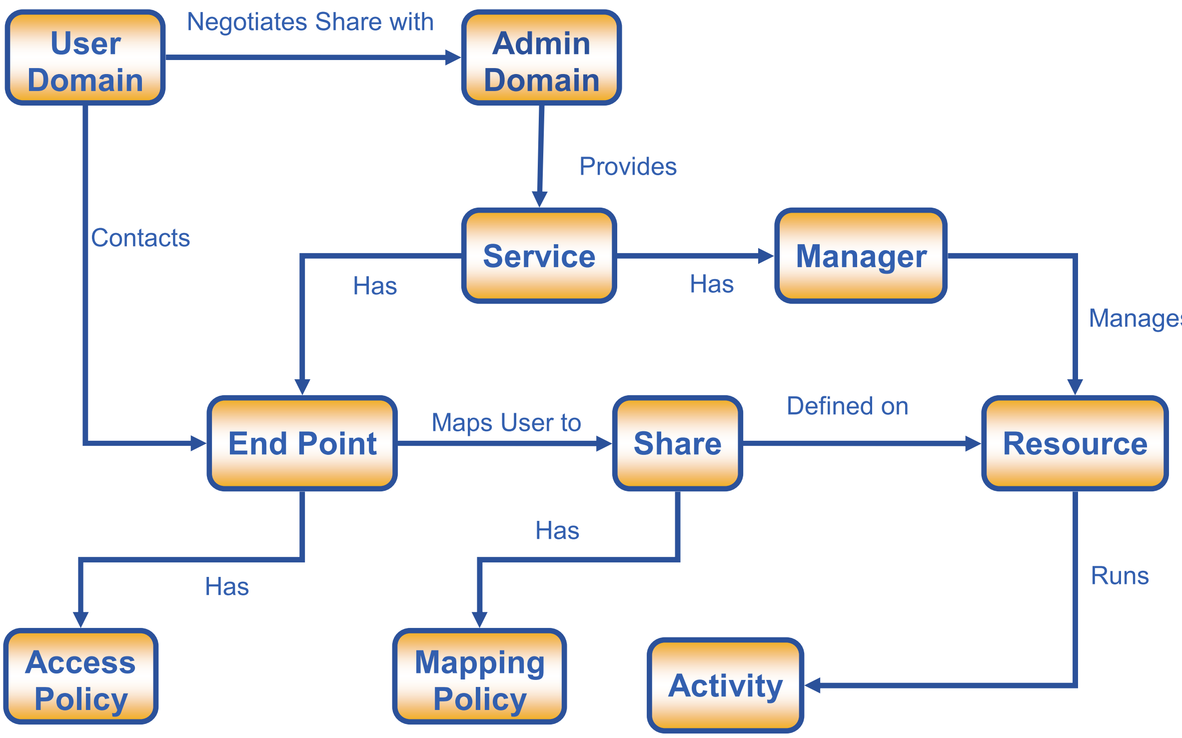


**Figure 1.** The main classes and relations in the GLUE 2.0 schema.

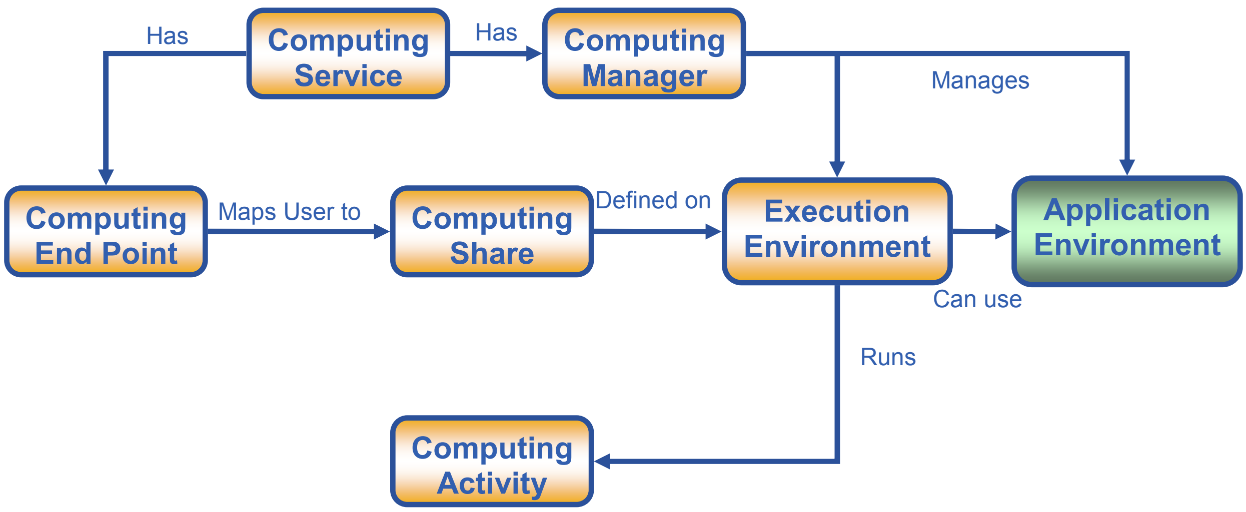


**Figure 2.** Specialised classes for computing services.

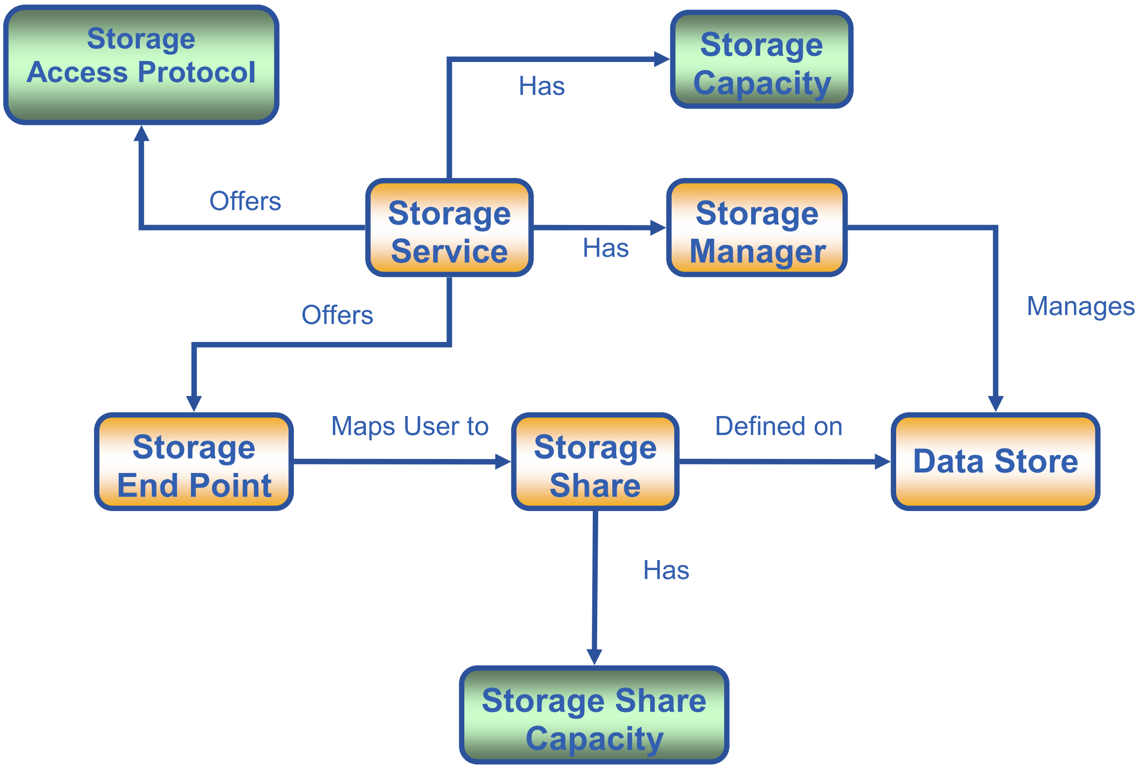


**Figure 3.** Specialised classes for storage services.

### 4.3. Changes from GLUE 1.3

The structure and naming in GLUE 2.0 appear to be somewhat different to GLUE 1.x, but some of this is just a result of changing terminology. The most significant such changes are listed in table 1. All existing use cases should still be satisfied by the new schema, and all the existing information is present in some form apart from things which are unused. There are also some additions driven by use cases from other Grid projects. These may not be published in the LCG/EGEE Grid; the intention is that a profile will be defined to define the usage of the schema within a particular Grid.

Some of the most significant structural changes from the 1.x schema are:

- There is a generic concept of a **Service** as a coherent grouping of **Endpoints**, **Managers** and **Resources**. **ComputingService** and **StorageService** are specialised variants of these, sharing a common structure as far as possible.
- There are generic representations for **Managers** (software) and **Resources** (hardware).
- All classes are extensible, with a multivalued string "OtherInfo" and/or key/value pairs (**Extension** classes).
- All objects have a globally unique ID attribute, hence arbitrary references and queries are possible.
- Many classes allow many-to-many relations with other classes. This allows more flexibility to reflect real-world configurations, but will also imply more complexity in implementation and usage.

- Some concepts have been made more generic and flexible than in the 1.x schema by making them separate classes rather than attributes of other classes; e.g., **Location**, **Contact**, **Policy**, **Benchmark** and **Capacity**. This makes it possible to have multiple instances of such objects with different types or for different purposes.
- Definitions of attributes have been made much more complete and rigorous, although experience suggests that ambiguities will still be found in practice. There are many more enumerated types, although these are not yet fully defined. There are also specifications for placeholder values, case sensitivity, and whether attributes are optional or mandatory.

**Table 1.** Equivalence between GLUE 1.x and GLUE 2.0 terminology. Terms in parentheses did not have an explicit representation in GLUE 1.x.

| GLUE 1 | GLUE 2 |
|---|---|
| Site | AdminDomain |
| (VO) | UserDomain |
| Element | Service |
| Service | Endpoint |
| AccessControlBaseRule | AccessPolicy, MappingPolicy |
| CE, VOView | ComputingManager, ComputingShare |
| Cluster, SubCluster | ExecutionEnvironment |
| (Job) | Activity |
| SA, VOInfo | StorageShare |

### 4.4. Schema representations
The 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 principle technology used in the LCG/EGEE Grid. At the time of writing the definition of these representations is not complete, but they are expected to be finalised by June 2009.

## 5. The impact of GLUE 2.0 on the LCG/EGEE Grid
There are several features of GLUE 2.0 which will constitute major improvements for the LCG/EGEE Grid. We have a large number of Grid services, hence the ability to publish everything in a common framework is highly desirable, and should enable the development of generic service discovery tools.

Our experience is that we frequently find new use cases which need new attributes, and modifying the schema is a rather slow process, so the fact that all classes can now be extended with new key/value pairs as well as arbitrary text tags is likely to be very useful.

We have a number of long-standing problems which were difficult or impossible to fix within the structure of the 1.x schema, which have been explicitly taken into account in the design of 2.0. None of these problems are show-stoppers, but many are annoying and awkward to work around. In particular we now have a storage schema designed for our current usage of the SRM protocol, and the computing schema separates the concepts of the Grid endpoint, batch system and queue/fairshare configuration which are entangled in the existing **CE** class.

Finally, we expect to gain from the move towards interoperability and standardisation that comes from moving GLUE into the OGF, especially as EGEE makes the transition to the new EGI [17] project. There is also a direct benefit to LCG as this brings the ARC [18] middleware into the GLUE framework for the first time, and ARC-based sites are a significant contributor to the worldwide LCG project. Also, as a Grid standard there is a possibility that we may get external implementations of the information providers at least for some services, rather than having to develop and maintain our own.

The following sections explain some of these changes in more detail.

### 5.1. Service discovery

For some time we have been working towards the publication of all Grid services in the LCG/EGEE Grid using the **GlueService**/**GlueServiceData** classes in the current schema, and this is now largely complete. However, this structure is rather limited in terms of the information which can be published. There are also some service discovery tools which allow users and other clients to obtain information about these services.

With the 2.0 schema we will have the possibility to publish more information about any service, although we will need to gain experience and initially it is likely that the existing information will be translated directly to the new format. However, even without extra information there is the immediate benefit that it brings the computing and storage schemas into the same framework, allowing generic query tools to be developed. There is already some work on this in the context of the OGF SAGA working group [19], which illustrates the potential benefits of working in the OGF community.

### 5.2. Storage systems

Over the course of 2008 the LCG/EGEE Grid introduced storage systems implementing version 2.2 of the SRM protocol. This has some major differences compared with the previous version 1.1, in particular in the use of space tokens, which identify portions of reserved space with specified properties for particular VOs. After some considerable discussion within the project we have found ways by which the 1.3 schema has allowed us to publish sufficient information to allow the use of the most important new features, but the situation is not ideal. The 2.0 schema has been explicitly designed with our current usage in mind, and hence will allow a much more accurate representation of current storage systems. The usage of the SRM protocol is still evolving and there are several outstanding issues so we may discover more problems in the future, but the flexibility and extensibility of the new schema should make it much easier to resolve them.

### 5.3. Resource accounting

In general, accounting is not a target use case for either GLUE 1.x or 2.0. The schema itself describes a snapshot of the current state and has no history, although it is possible to extract information at regular intervals and store it in a separate database. Fine-grained information, e.g. at the level of individual jobs or files, might lead to a volume of information which exceeds the capacity of the information system, although GLUE 2.0 does have the **Activity** class which can potentially be used for such information. There is also no explicit provision for the encryption or signing of potentially sensitive user-level data, and the BDII information system technology used in the LCG/EGEE Grid currently has all information world-readable.

However, while we do have an independent accounting system for the use of computing resources there is currently no similar solution for storage accounting. Additionally, during 2008 LCG has developed requirements for high-level management monitoring of both computing and storage resources, in terms of the total installed capacity at each Grid site and the overall allocation to and usage by each VO. These requirements are currently satisfied using information published according to the 1.3 schema [20]. However, this is not ideal as the schema structure makes it difficult to represent all the desired information, and this situation will be much improved with the 2.0 schema. For example, the intention is to measure computing resources using a new benchmark; GLUE 1.x only allows for the obsolete SpecInt2000 and SpecFloat2000 benchmarks, whereas the 2.0 schema has a separate **Benchmark** class which can publish the CPU power using arbitrary benchmarks. Similarly there is explicit provision for dealing with such issues as internal scaling of CPU times in a batch system, distinguishing between installed and available capacity and specifying the sharing between VOs, which are not supported by the 1.x schema.

For general storage accounting there is no explicit support in the 2.0 schema (in particular there is currently no definition of a **StorageActivity** class). However, the improved description of SRM 2.2 space tokens will enhance the accuracy of the current coarse-grained storage accounting system.

### 5.4. Implementation plan

With the schema definition now finalised, we can start deploying the schema in the LCG/EGEE Grid. However, this is expected to be a fairly long process as all changes have to be made in a way which allows a fallback to the existing system. The principle steps will be:

- The LDAP representation of the schema will be defined, and deployed to all BDII servers in parallel with the existing schema. This is not a disruptive change and should be complete by mid-2009.
- Information providers to publish information according to the new schema must be written and deployed, again in parallel with the existing system. This is likely to be rolled out gradually as the providers are written; we aim to have a substantial number of services published by the end of 2009, with publication of the more complex services being completed during 2010.
- 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 which have not upgraded to the new publishers. Experience suggests that this will be a slow process, with a possible target date being the end of 2011 for the major clients.
- At some point it should 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 migration process will be rather slow, and may be complicated by the expected transition from EGEE to EGI during 2010. However, for the reasons described above we believe that the potential gains are sufficient to justify making the transition.

## 6. Conclusions

The GLUE schema has developed over eight years of use by the EDG, LCG and EGEE projects, as well as other Grids. It has proved sufficient to allow many users to submit large numbers of jobs, manage data and monitor the Grid. There are therefore no showstoppers, but there are many rough edges and known limitations.

The time is right for a major new schema version that incorporates our experience, and also collects input from many other Grids. Framing this process within the OGF allows this work to reach a worldwide audience, which should help with interoperability and encourage other Grids to participate. GLUE 2.0 should cover all current use cases for LCG/EGEE, as well as allowing things we are unable to do using the 1.3 schema. It will also be much more flexible for the new uses that have not yet been anticipated but which experience suggests are likely to occur. The implementation and deployment of the new schema will start in 2009, but the transition process will likely take several years.

## 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 (10pp)
Andreozzi S, Burke S, Field L and Konya B 2008 Towards GLUE 2: evolution of the computing element information model *J. Phys:. Conf. Ser.* **119** 062009 (6pp)
[2] GLUE working group http://forge.ogf.org/sf/projects/glue-wg
[3] 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
[4] LCG: LHC Computing Grid http://www.cern.ch/lcg
[5] EGEE: European Grid for E-sciencE http://www.eu-egee.org
[6] OSG: Open Science Grid http://www.osg.org
[7] Nordic Grid Facility http://www.ndgf.org
[8] BDII: https://twiki.cern.ch/twiki//bin/view/EGEE/BDII
[9] EU DataGrid Project http://cern.ch/eu-datagrid
[10] EU DataTAG Project http://cern.ch/datatag
[11] iVDGL: International Virtual Data Grid Laboratory http://www.ivdgl.org
[12] The Globus Project http://www.globus.org
[13] 2003 GLUE Schema Specfication - Version 1.1 http://glueschema.forge.cnaf.infn.it/Spec/V11
[14] Andreozzi S, Burke S, Field L, Fisher S, Konya B, Mambelli M, Schopf J, Viljoen M and Wilson A 2005 GLUE Schema Specifiation - Version 1.2 http://glueschema.forge.cnaf.infn.it/uploads/Spec/GLUEInfoModel_1_2_final.pdf
[15] 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 Specifiation - 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
[16] Storage Resource Management Working Group http://sdm.lbl.gov/srm-wg/
[17] EGI: European Grid Initiative http://web.eu-egi.eu/
[18] ARC middleware: http://www.nordugrid.org/middleware/
[19] OGF SAGA Working Group: http://www.ogf.org/gf/group_info/view.php?group=saga-wg
[20] Burke S,Cowan G, Donno F, Field L, Jensen J, Jouvin M, Marques Coelho Dos Santos M, Magnoni L, Millar P, Shih J, Templon J, Traylen S, Trompert R, Schiaua C, Van Eldik J, Zappi R, Holzman B and Bockelman B 2009 Usage of Glue Schema v1.3 for WLCG Installed Capacity information https://twiki.cern.ch/twiki/pub/LCG/WLCGCommonComputingReadinessChallenges/WLCG_GlueSchemaUsage-1.8.pdf