



INFN/TC-04/16
30 Settembre 2004

GLUE SCHEMA IMPLEMENTATION FOR THE LDAP DATA MODEL*

Sergio Andreatozzi

INFN-CNAF, Viale Berti Pichat 6/2, 40127 Bologna, Italy
email: sergio.andreatozzi@cnafe.infn.it

Abstract

An important aspect of the interoperability among different Grid systems is the definition of a common schema for the representation of Grid resources. This is vital for sharing the status and availability of resources at the virtual level. The GLUE schema is designed to fulfil this need by providing a conceptual schema of the relevant resources. This paper presents a mapping of this schema into the LDAP data model. It is used in the Globus MDS version 2.

Published by SIS-Pubblicazioni
Laboratori Nazionali di Frascati

* This research was partially funded by the IST Program of the European Union under grant IST-2001-32459.

1 Introduction

The Grid Laboratory Uniform Environment (GLUE) [3] is a collaboration effort between a number of European and American Grid related projects that aims at producing a set of building blocks for the interoperability among different Grid systems. The GLUE Schema is the proposal for a common representation of Grid resources, their state and functionalities. It is conceived to be used as a base schema for the Grid Information Service (GIS) for both discovery and monitoring purposes. The conceptual design has been described by means of UML class diagrams supported by descriptive documents. The purpose of this approach is to provide a meta-schema from which mappings to different data models and technologies can be derived. This document describes the mapping of both computing [6] and storage [7] resources in a directory-based data model such is the LDAP (Lightweight Directory Access Protocol) [10]. The target GIS is the Globus Monitoring and Discovery Service (MDS), version 2 [8,9]. The latest available release is 2.4.3 and it is based on the OpenLDAP version 2.0.22 [5].

2 Overview of the GLUE Schema

The latest specification of the GLUE schema is numbered 1.1 and provides representations for computing and storage resources. As regards computing resources, the defined entities are [6]: ‘Computing Element’ (CE), entry point in a batch system with a 1-to-1 relationship with a queue; ‘Host’, a computer system; ‘Cluster’, a set of hosts managed by a batch system; ‘SubCluster’, and abstract concept representing the common characteristics of a subset of hosts in a cluster; given a set of characteristics, the cluster can be divided into disjointed set of hosts, each one described by a SubCluster.

Among the storage resources [7], the main entities are: ‘Storage Service’ (SE), a uniquely identified grid service that manages storage extent in term of storage spaces; ‘Storage Space’ (SA), portion of a logical storage extent with an homogenous set of policies (typically used to assign space to VOs); ‘Data Access Protocol’, protocol to be used to access data in a Storage Space; ‘Storage Library’ (SL), a machine providing for storage services.

3 The LDAP schema

The current UML class diagram shows each resource as composed of a set of classes, each one capturing a certain aspect or functional part of the resource. For instance, the CE is composed of the following classes: Info, State, Policy and AccessControlBase. Stated this, both *attribute* and *objectclass* names have a common structure: the structure of an

objectclass name is `prefix + resource (+ component)`; the structure of an attribute class name is `prefix + resource (+ component) + attribute`.

The prefix is always `Glue` and defines a namespace for the schema in order to avoid collisions with names defined in other contexts (the LDAP naming is flat). For instance, considering the `CE` class, its corresponding objectclass name is `GlueCE`, the attribute `UniqueID` is mapped to the LDAP attribute `GlueCEUniqueID`; the component class `State` has the corresponding LDAP objectclass `GlueCEState`, while the attribute `RunningJobs` is mapped to the LDAP `GlueCEStateRunningJobs`.

4 The Directory Information Tree

A Directory Information Tree is the structure of the hierarchy of data contributed by several LDAP servers (in the MDS terminology `GRIS` and `GIIS`). Each node of this tree is called entry. Each entry has a Relative Distinguished Name (RDN, a couple `attribute-Name=attributeValue`) that is required to be unique among all its siblings. “The concatenation of the relative distinguished names of the sequence of entries from a particular entry to an immediate subordinate of the root of the tree forms that entry’s Distinguished Name (DN), which is unique in the tree” [10]. Each entry can store one or more objectclasses. An instance of an objectclass requires a set of mandatory attributes and a set of optional attributes. Since an objectclass instance does not have a name, one entry can contain only one instance per objectclass type.

4.1 Base DIT design in the Globus MDS

The Globus MDS version 2 provides two main components, the `GRIS` and the `GIIS`. The `GRIS` provides an LDAP tree for publishing data about a specific set of Grid resources. Entries related to these resources have the ancestor `mds-vo-name=local,o=grid` (this is called suffix and is the standard choice for `GRIS`s of the Globus MDS) [9]. The `GIIS` accepts registration from `GRIS`s or `GIIS`s and performs aggregation and caching of the provided data. Its suffix is `mds-vo-name=aName,o=grid` and `aName` must be different than `local` (typically it is filled with organization name or organizational unit name). A `GIIS` provides a single point of query for all the registered MDS components.

The DIT structure evolves by adding `GRIS` or `GIIS` components to the hierarchy. Let us consider the following example in order to understand how the global tree is constructed. The `CNAF` site provides two Grid elements, a `CE` and a `SE` located respectively at the `edt001.cnaf.infn.it` and `edt004.cnaf.infn.it` hosts. Each of them runs an MDS `GRIS` whose Uniform Resource Identifiers (URIs) are respectively:

1. `ldap://edt001.cnaf.infn.it:2135/mds-vo-name=local,o=grid`
2. `ldap://edt004.cnaf.infn.it:2135/mds-vo-name=local,o=grid`

By configuring an MDS GIIS, these resources can be aggregated on a site basis. For instance, a GIIS can be enabled on the first machine and the URI could be `ldap://edt001.cnaf.infn.it:2135/mds-vo-name=CNAF,o=grid`. When a GRIS registers to the GIIS, all the GRIS entries are also accessible from the GIIS. All the GRIS entries rooted in `mds-vo-name=local,o=grid`, will be rooted to the GIIS suffix. The site GIIS can then register to a national GIIS, e.g.: the INFN GIIS, which URI could be: `ldap://grid.infn.it:2135/mds-vo-name=INFN,o=grid`. The resulting DIT in each MDS component is shown in figure 1.

4.2 DIT design for the GLUE Schema

Top level entities are Services and Systems. Regarding computing resources, each Computing Element is an independent entry. At the same level, a cluster entry is present. The cluster entry is the root for a set of host-related entries. The cluster entry can have one or more subcluster children entries summarizing the homogenous characteristics of subsets of homogenous osts. Further, the cluster entry can have one or more hosts entries with the respective children entries referring to local file system, remote file system and network adapter info (see Figure 2). Regarding storage resources, for each provided system, the DIT shows one entry for the Storage Library and one or more entries for the provided Storage Services. Each Storage Service manages a set of Storage Spaces whose files can be accessed by choosing one of the advertised Data Access Protocols (see Figure 3).

4.3 Expressing relationships between Computing and Storage Services

For brokering issues, it is important to be able to express relationships between services, i.e., between CEs and SEs. Since this relationship has a many-to-many multiplicity, the chosen strategy has been the following (see Figure 4):

- for a certain CE, a group level entry lists all the related SEs
- for each SE, a single level entry shows the attributes characterizing the specific CE-SE relationship

At the moment, the only important attribute is the mount point that a running job should use to locally access a Storage Space managed by the SE.

4.4 Expressing relationships between LDAP entries

Entries in an LDAP tree can be accessed by their distinguished names or by using declarative queries. The basic LDAP query is composed of four parts:

1. the root of the search (-b option)
2. the scope of the search (-s option, can be base, one or sub)
3. a filter composed of a conjunction and/or disjunction of predicates
4. a list of attributes to be returned

Components from 1 to 3 are used to select the entries that will be returned. Component 4 is used to select, among the matching entries, which part should be returned to the user. An interesting option provided by the Extensible Matching is the ability to select entries on the base of constraints applied to its DN. Unfortunately, this is implemented in OpenLDAP only in 2.1.x version, i.e. this is not available with MDS. This led to the need of finding out a way to answer to: "I want to know all the Storage Spaces managed by a certain Storage Service". To accomplish this task, the attribute `GlueChunkKey` has been introduced. The DN is a unique key of a certain entry and is a concatenation of RDN's. Each RDN is therefore a chunk key, this is the reason for the attribute name. By specifying in each entry related to a Storage Space the attribute

```
GlueChunkKey: GlueSEUniqueID=edt004.cnaf.infn.it
```

it is possible to express the requested query (see Figure 5):

```
ldapsearch -h edt004.cnaf.infn.it -p 2135 -x -b "mds-vo-name=local,o=grid"
'(&(objectclass=GlueSA)(GlueChunkKey=GlueSEUniqueID=edt004.cnaf.infn.it))'
```

With the extensible matching support it could be expressed as follows:

```
ldapsearch -h edt004.cnaf.infn.it -p 2135 -x -b "mds-vo-name=local,o=grid"
"(&(objectclass=GlueSA)(GlueSEUniqueID:dn:=edt004.cnaf.infn.it))"
```

Another issue is to express constraints on attributes part of an entry not part of the DN of the entry of interest. For example, "I want to know all the CEs provided by the cluster `edt001.cnaf.infn.it`". To accomplish this task, the `GlueForeignKey` attribute has been introduced. By specifying in each CE entry the attribute:

```
GlueForeignKey:GlueClusterUniqueID=edt001.cnaf.infn.it
```

it is possible to express the requested query as follows (see Figure 5):

```
ldapsearch -h edt001.cnaf.infn.it -p 2135 -x -b "mds-vo-name=local,o=grid"
"(&(objectclass=GlueCE)(GlueForeignKey=GlueClusterUniqueID=edt001.cnaf.infn.it))"
```

4.5 Querying the Information Tree

In this section, a number of example of LDAP queries and results to a top-level GIIS of the INFN Grid are provided.

4.5.1 CE: ID accessible by the CMS VO

```
QUERY:
ldapsearch -h ibm140.cnaf.infn.it -p 2170 -x -LLL -b "mds-vo-name=local,o=grid"
'(&(objectclass=GlueCE)(GlueCEAccessControlBaseRule=VO:CMS))' GlueCEUniqueID

ANSWER:
dn: GlueCEUniqueID=lxn1184.cern.ch:2119/jobmanager-lcglsf-grid,mds-vo-name=cernlsg2,mds-vo-name=local,o=grid
GlueCEUniqueID: lxn1184.cern.ch:2119/jobmanager-lcglsf-grid

dn: GlueCEUniqueID=grid002.pi.infn.it:2119/jobmanager-lcgpbs-long,Mds-Vo-name=infn-pisa,mds-vo-name=local,o=grid
GlueCEUniqueID: grid002.pi.infn.it:2119/jobmanager-lcgpbs-long

dn: GlueCEUniqueID=gridba2.ba.infn.it:2119/jobmanager-lcgpbs-long,Mds-Vo-name=infn-bari,mds-vo-name=local,o=grid
GlueCEUniqueID: gridba2.ba.infn.it:2119/jobmanager-lcgpbs-long
```

4.5.2 CE: variables of the execution environment

The Computing Element is characterized by attributes belonging to the policies for the access to the available executions environments. In order to have an access-related information (authorization) and an execution environment information (related to the subcluster), two queries are needed. For each CE, query 1 returns all the references to the clusters that are accessible by them for the CMS VO. Query 2 returns the characteristics of the execution environment of a specific cluster.

```
QUERY 1:
ldapsearch -h ibm140.cnaf.infn.it -p 2170 -x -LLL -b "mds-vo-name=local,o=grid"
'(&(objectclass=GlueCE)(GlueCEAccessControlBaseRule=VO:CMS))' GlueCEUniqueID GlueForeignKey

ANSWER 1:
dn: GlueCEUniqueID=virgo-ce.roma1.infn.it:2119/jobmanager-lcgpbs-infinite,
Mds-Vo-name=infn-roma1-virgo,mds-vo-name=local,o=grid
GlueCEUniqueID: virgo-ce.roma1.infn.it:2119/jobmanager-lcgpbs-infinite
GlueForeignKey: GlueClusterUniqueID=virgo-ce.roma1.infn.it
```

```
QUERY 2:
ldapsearch -h ibm140.cnaf.infn.it -p 2170 -x -LLL -b "mds-vo-name=local,o=grid"
'(&(objectclass=GlueSubCluster)(GlueChunkKey=GlueClusterUniqueID=virgo-ce.roma1.infn.it))'
```

```
ANSWER 2:
dn: GlueSubClusterUniqueID=virgo-ce.roma1.infn.it,GlueClusterUniqueID=virgo-ce.roma1.infn.it,
Mds-Vo-name=infn-roma1-virgo,mds-vo-name=local,o=grid
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_1
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH
GlueHostApplicationSoftwareRunTimeEnvironment: ROMA1-VIRGO
GlueHostApplicationSoftwareRunTimeEnvironment: VIRGO-1.0
```

4.5.3 SA: ID and available space accessible by the CMS VO

```
QUERY:
ldapsearch -h ibm140.cnaf.infn.it -p 2170 -x -LLL -b "mds-vo-name=local,o=grid"
'(&(objectclass=GlueSA)(GlueSAAccessControlBaseRule=CMS))' GlueSARoot GlueSAStateAvailableSpace

ANSWER:
dn: GlueSARoot=cms:cms,GlueSEUniqueID=gridba6.ba.infn.it,Mds-Vo-name=infn-bari,mds-vo-name=local,o=grid
GlueSARoot: cms:cms
GlueSAStateAvailableSpace: 291009000

dn: GlueSARoot=cms:cms,GlueSEUniqueID=grid2.fe.infn.it,Mds-Vo-name=infn-ferrara,mds-vo-name=local,o=grid
GlueSARoot: cms:cms
GlueSAStateAvailableSpace: 21304676
```

5 Conclusions

In this paper, the mapping of the conceptual modeling of Grid resources given with the GLUE Schema into an LDAP data model was presented. It has been validated in the context of the European DataTAG project [2]. Later, it has been integrated in a number of Grid middlewares relying on the Globus MDS version 2 (e.g., European DataGrid [1], LHC Computing Grid [4]).

References

- [1] *The DataGrid project*, <http://www.edg.org>.
- [2] *The DataTAG project*, <http://www.datatag.org>.
- [3] *Grid Laboratory Uniform Environment (GLUE) Collaboration*, <http://www.hicb.org/glue/glue.htm>.
- [4] *LHC Computing Grid project*, <http://www.cern.ch/lcg/>.
- [5] *OpenLDAP*, <http://www.openldap.org>.
- [6] *GLUE Schema for the Computing Element, Version 1.1*, <http://www.cnaf.infn.it/~sergio/datatag/glue/v11/CE>, 2003.
- [7] *GLUE Schema for the Storage Element, Version 1.1*, <http://www.cnaf.infn.it/~sergio/datatag/glue/v11/SE>, 2003.
- [8] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, *Grid Information Services for Distributed Resource Sharing*, Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), San Francisco, CA, USA, Aug 2001.
- [9] The Globus Team, *Globus Toolkit 2.2 MDS Technology Brief, Draft 4*, Jan 2003.
- [10] W. Wahl, T. Howes, and S. Kille, *Lightweight Directory Access Protocol v.3, RFC 2251, IETF*, Dec 1997.

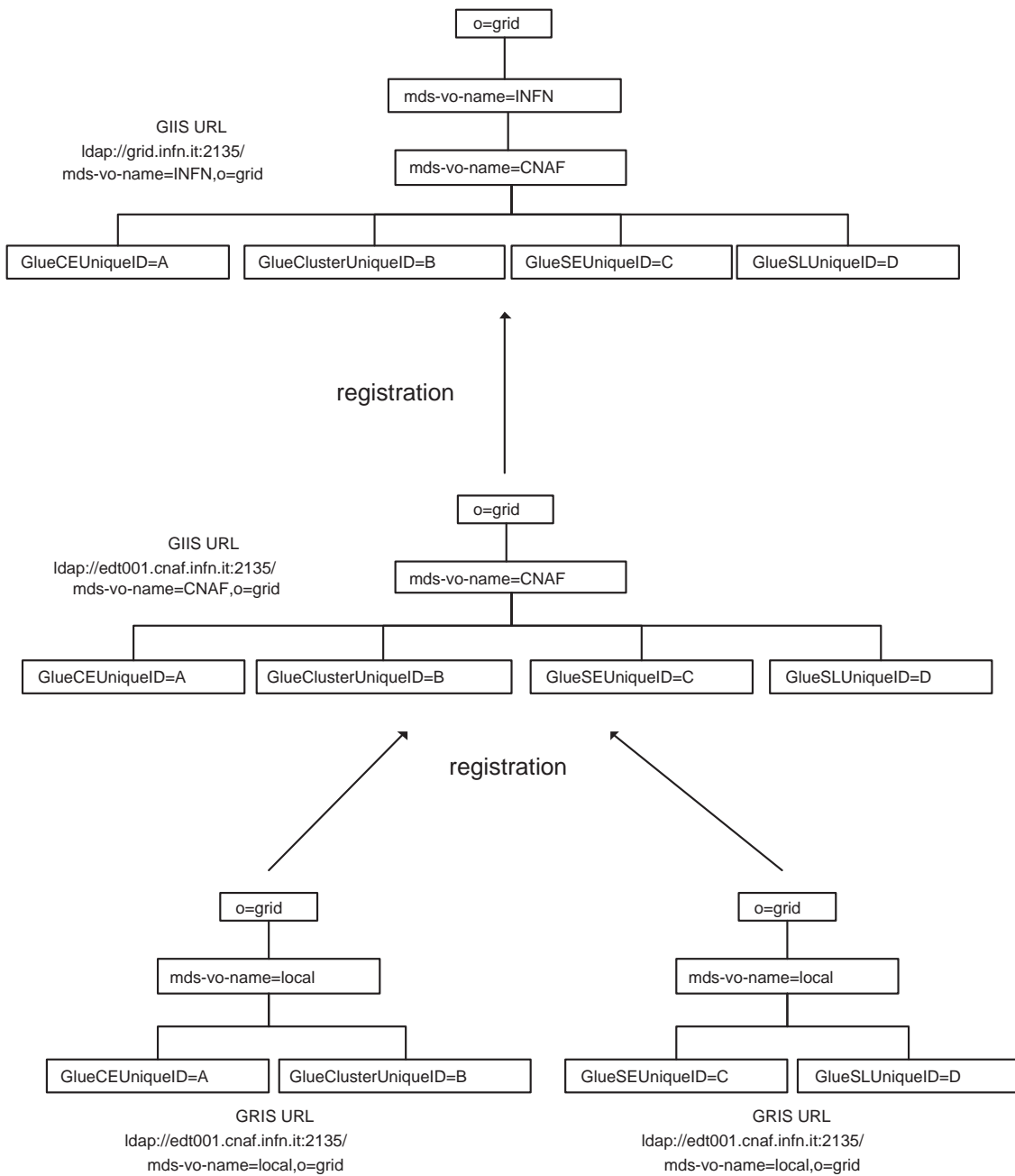


Figure 1: GRIS-GIIS and GIIS-GIIS registration examples

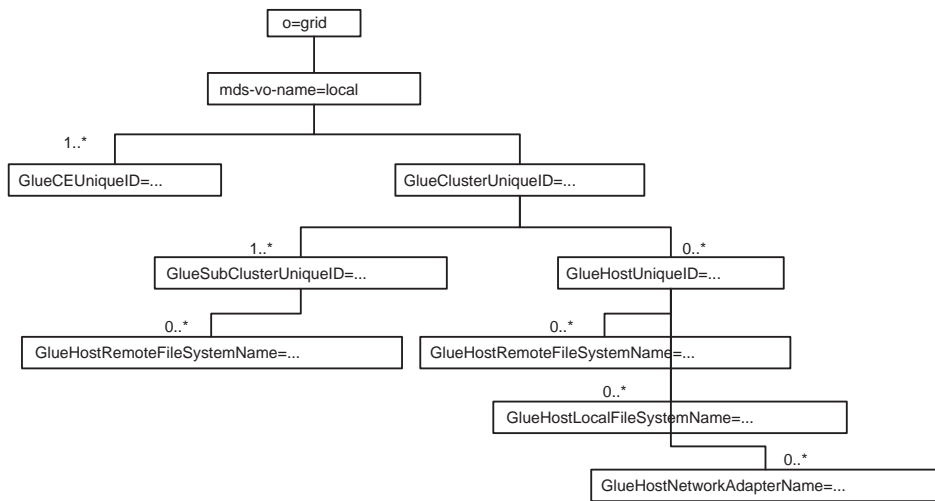


Figure 2: DIT for the Computing Resources

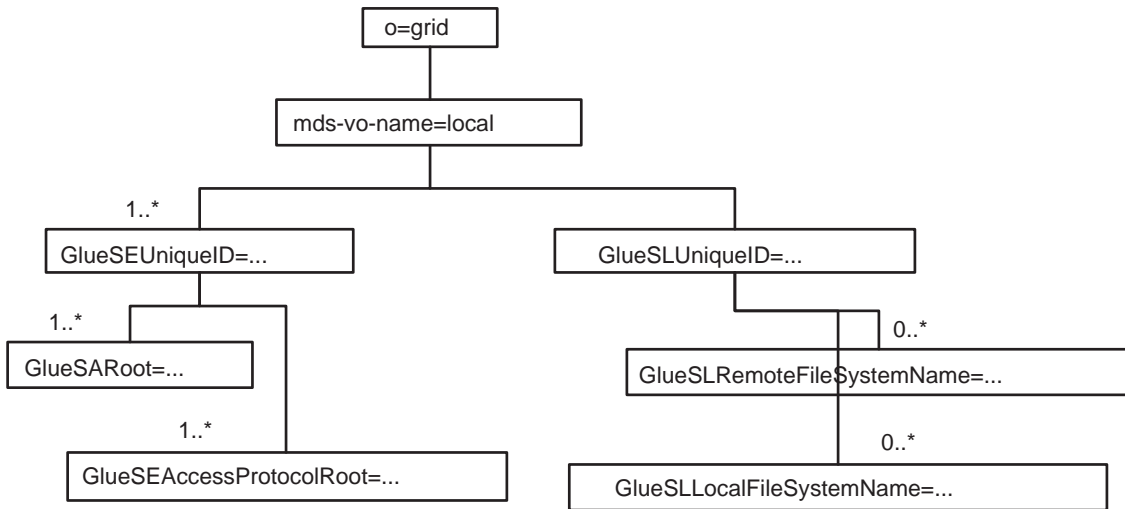


Figure 3: DIT for the Storage Resources

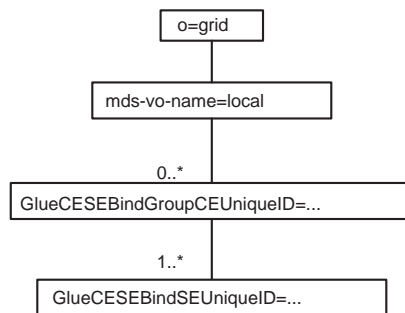


Figure 4: DIT for the CE-SE relationship

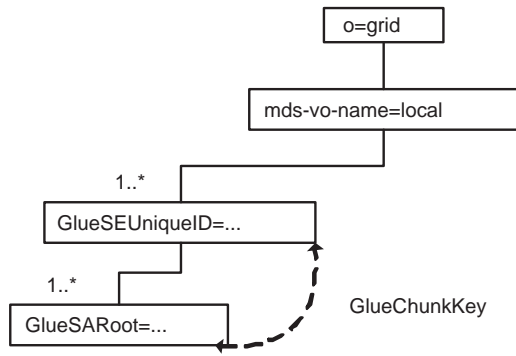


Figure 5: Example of GlueChunkKey attribute use

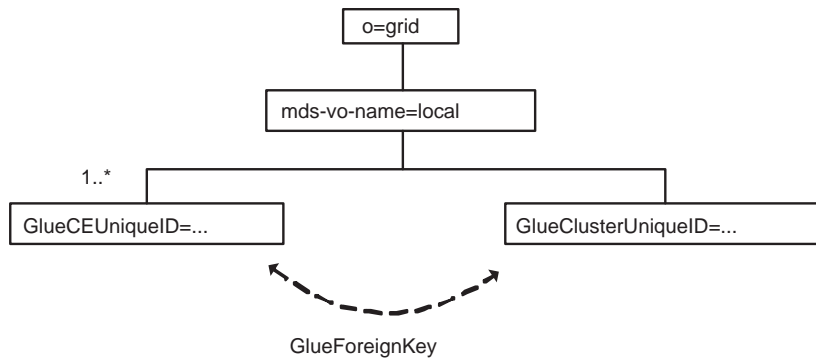


Figure 6: Example of GlueForeignKey attribute use