GWD-R, GWD-I or GWD-C
Authors:

Sergio Andreozzi* (editor), INFN
Stephen Burke, RAL
Felix Ehm, CERN
Laurence Field*, CERN
Gerson Galang, ARCS
David Horat, CERN
Balazs Konya*, Lund University
Maarten Litmaath, CERN
Paul Millar, DESY
JP Navarro, ANL
*co-chairs

GLUE WG
http://forge.ogf.org/sf/sfmain/do/viewProject/projects.glue-wg

April 23, 2009

GLUE v. 2.0.1 – Reference Realizations to SQL Schema

Status of This Document

This document provides information to the Grid community regarding the Relational Schema in SQL realization of the GLUE information model (v.2.0). Distribution is unlimited. This implementation is derived from the specification document "GLUE Specification v. 2.0.1", April 23, 2009. This document is a draft.

Abstract
The GLUE specification is an information model for Grid entities described in natural language enriched with a graphical representation using UML Class Diagrams. This document presents a realization of this information model as XML Schema, LDAP Schema and SQL Schema.

Contents

1. Introduction

The GLUE 2.0 Information model defined in [glue-2] is a conceptual model of Grid entities. In order to be adopted by Grid middlewares, a realization in terms of a concrete data model is needed.

This document provides the normative realization of the GLUE 2.0 conceptual model in terms of an Relational schema in SQL. The approach followed to map the entities and relationships in the conceptual model to the concrete data model are also described.

2. Notational Conventions

The key words 'MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" are to be interpreted as described in RFC 2119 (see http://www.ietf.org/rfc/rfc2119.txt).

3. SQL Realization

3.1 Approach

There are many approaches to realize the GLUE conceptual model as an SQL implementation. The GLUE SQL rendering maps an entity in the GLUE information model to a specific table. As a result there is one to one correspondence between GLUE SQL tables and GLUE entities.

The SQL language taken as reference is SQL-92, which is defined in [sql-92], because it is the one which is implemented by most database servers.

3.2 Inheritance

Inheritance has been treated in SQL so every entity level is been contained in a table. E.g.: For the GLUE entity UserDomain which inherits from Domain, there will be an entry in the UserDomain table and the Domain table. In order to achieve this, all abstract entities have also been implemented as tables. The only exception for this behavior is the "Entity" entity, which fields have been directly implemented in each of its superclasses.

3.3 Multivalued attributes

Since SQL itself does not allow multivalued attributes, other approaches must be taken. In our case, we decided between three different possibilities, all of which involve creating new tables:

1. Create a new table for each multivalued attribute. The name of this table will contain both the entity name and the multivalued attribute name. Its fields will be "Id" and "Value". "Id" will be a foreign key to the "Id" of the original entity table. The primary key of this table will be the combination of "Id" and "Value".
2. Create a new table for each table which contains multivalued attributes. The name of this table will contain the entity name and the suffix "_MVA". Its fields will be "Id", "Type" and "Value". "Id" will be a foreign key to the "Id" of the original entity table. "Type" will be a character field which will contain the name of the attribute. The integrity of the field "Type" will be held by other means. The primary key of this table will be the combination of "Id", "Type" and "Value".
3. Create one table overall. The name of this table will be "MVA". Its fields will be "Id", "Table", "Type" and "Value". "Id" will be a foreign key to the "Id" of the original entity table. "Table" will be a character field which will contain the name of the table. "Type" will be a character field which will contain the name of the attribute. The integrity of the fields "Table" and "Type" will be held by other means. The primary key of this table will be the combination of "Id", "Table", "Type" and "Value".

Of the three possibilities, the second one was chosen. Although the first one is the most proximately to the GLUE 2.0 Specifications, the second one is the best in terms of optimization purposes. Thus, the second option was chosen.

3.4 Data types

For the implementation of the different data types, we have decided to just use two different types of the standard SQL-92:

- **VARCHAR(255)**
- **INTEGER**

"INTEGER" will be used for types UInt32 and UInt64 of the original GLUE 2.0 Specification and "VARCHAR(255)" will be used for every other type.

This also means that data type integrity will not be held in the LDAP implementation itself, but must be ensured by other means.

4. Security Considerations

Using SQL to implement GLUE 2.0 Specifications raises several considerations specially in the field of data integrity.

SQL can not ensure all data types referred in the GLUE 2.0 Specifications, thus these must be ensured by other means.

5. Author Information

Sergio Andreozzi, INFN

Stephen Burke, RAL

Felix Ehm, CERN

Laurence Field, CERN

Gerson Galang, ARCS

David Horat, CERN

Balazs Konya, Lund University

Maarten Litmaath, CERN

Paul Millar, DESY

JP Navarro, ANL

6. Contributors & Acknowledgements

We gratefully acknowledge the contributions made to this document.

7. Intellectual Property Statement

such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

8. Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

9. Full Copyright Notice

10. References

[glue-wg] The GLUE Working Group of OGF, *https://forge.gridforum.org/sf/projects/glue-wg*
[glue-usecases] GLUE 2.0 Use Cases (early draft), *https://forge.gridforum.org/sf/go/doc14621*
[glue-2] GLUE 2.0 Specification, *https://forge.gridforum.org/sf/go/doc146239*
[ogf-ns] Standardised Namespaces for XML infosets in OGF,
http://www.ogf.org/documents/GFD.84.pdf
[xsd-oe] XForms 1.0. Open Enumeration.
http://www.w3.org/TR/2002/WD-xforms-20020118/slice6.html#model-using-openenum
[xsd-ap] Advanced XML Schema Patterns for Databinding Version 1.0,
http://www.w3.org/TR/xmlschema-patterns-advanced/#group-Unions
[sql-92] ISO/IEC 9075:1992 - Database Language SQL (SQL-92),
http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt