GWD-R.xx
SAGA-RG
Version: 0.1

Andre Merzky (CCT/LSU)

January 15, 2008

SAGA API Extension: Transactions

Status of This Document

This document provides information to the grid community, proposing a standard for a transaction extension to the Simple API for Grid Applications (SAGA). This document is intended to be used as input to the definition of language specific bindings for this API extension, and as reference for implementors of these language bindings. Distribution of this document is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2008). All Rights Reserved.

Abstract

This document provides information to the grid community, proposing a standard for a transaction extension to the Simple API for Grid Applications (SAGA). As such it depends upon the SAGA Core API Specification [**?**]. It also refers to the notion of transactions as defined by the set of DAIS specfications [**?**, **?**, **?**].

Contents

1 Introduction

The 'Simple Api for Grid Applications' Working Group (SAGA-WG) in OGF strives for a uniform OGF API, which is supposed to cover all high level application programming aspects of the OGF standardization landscape. The SAGA Core API specification is also a 'Proposed OGF Standard'.

The SAGA API is modular, and consists of (i) a set of nonfunctional packages (the SAGA Look & Feel), and (ii) a set of functional packages, which provide the respective programming paradigms to the Grid application programmers. The SAGA Core API includes, amongst others, a package for job submission and management. That package builds upon the experiences from various OGF groups and from other Grid APIs (BES, DRMAA, JSDL, GAT, CoG, GridSphere, ...).

This document describes an additional Look & Feel package, which is supposed to allow transaction semantics for functional packages.

2 SAGA Transaction API

2.1 Introduction

2.2 Specification

```
package saga.transaction
{
  // transaction state
  {
    New    = 0,
    Open   = 1,
    Closed = 2,
    Failed = 3
  }

  // transaction initialization mode
  enum mode
  {
    Automatic = 0,
    Manual    = 1
  }

  // transaction isolation policy
  enum policy
  {
    ReadUncommitted = 0,
    ReadCommitted   = 1,
    RepeatableRead  = 2,
    Serialisable    = 3
  }

  interface transactions : extends saga::async
  {
    transaction_init      (in  mode   m,
                           in  policy p);
    transaction_rollback  (void);
    transaction_commit    (void);
    transaction_get_state (out state  s);
  }
}
```

2.3 Enum `state`

SAGA objects which support transactions are stateful, i.e. they are in exactly one transactional state at any point in time. The state diagram is extremely simple, and is shown in Figure 1. A new object instance has a transactional state of `New`. Once initialized with calling `transaction_init()`, the transaction is `Open`. After calling either `transaction_commit()` or `transaction_rollback()`, the object transaction goes back into the `Commited` or `RolledBack` state, respectively. From there, another `transaction_init` leads back to the `New` state, and a new transaction starts.

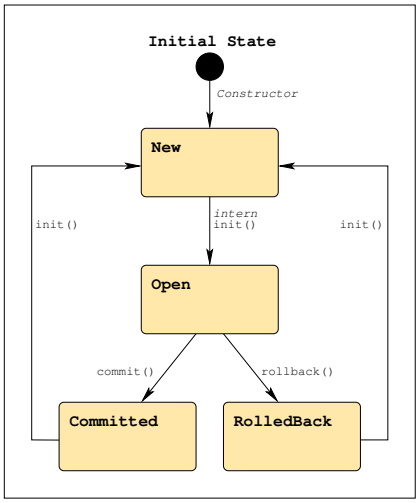

Figure 1: The SAGA transaction state model.

2.3.1 Examples

3 Conclusions

As stated in the Introduction 1, this document serves three different purposes. This conclusions provides a number of observations in respect to these points – more discussions are needed to finalize the conclusions.

Prove that a DRMAA rendering in SAGA is possible

In general, a DRMAA rendering within SAGA seems very well possible, and allows to maintain the DRMAA semantics. A number of differences, mostly in the API Look & Feel, must be noted however:

- DRMAA Exceptions do not map 1:1 to SAGA exceptions. The SAGA API does not, however, allow to introduce new exceptions in packages. Some care must thus be taken to prvide a reasonable mapping from DRMAA exceptions to SAGA exceptions.
- The DRMAA API is procedural, the SAGA API is object oriented. The mapping of procedural APIs to object oriented representations is inherently difficult and can be done in multuitude of ways. The presented mapping stays true to the mapping procedure used for the original SAGA job package, and has the implication that a number of job management methods are not performed on the job service handle (as in DRMAA), but on the job instance itself (as normal in SAGA).

allow for an easy comparison between the original SAGA job API and the DRMAA API

pave the path for a DRMAA API package in SAGA for the planned DRMAA version 2.0

4 Intellectual Property Issues

4.1 Contributors

This document is the result of the joint efforts of several contributors. The authors listed here and on the title page are those committed to taking permanent stewardship for this document. They can be contacted in the future for inquiries about this document.

Andre Merzky
andre@merzky.net
Center for Computation and Technology
Louisiana State University
216 Johnston Hall
70803 Baton Rouge
Louisiana, USA

In particular, the document build heavily on the specifications of the OGF GridRPC Working Group – we want to thank Eddy Caron, Craig Lee, Hidemoto Nakata and Yusuke Tanimura for their input and cooperation.

4.2 Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of 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.

4.3 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.

4.4 Full Copyright Notice

Copyright (C) Open Grid Forum (2007). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.