SAGA API Extension: Database Access and Integration

Status of This Document

This document provides information to the grid community, proposing a standard for a DAIS 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.

Abstract

This document provides information to the grid community, proposing a standard for a DAIS extension to the Simple API for Grid Applications (SAGA). As such it depends upon the SAGA Core API Specification [3], und upon the set of DAIS specfications [1, 4, 2].

Contents

1   Introduction

The 'Database Access and Integration Services Working Group' (DAIS-WG) in OGF has defined a set of service descriptions which provide data model independent interfaces to data bases. Several implementations of the DAIS services exist and are in use, and the specification has completed the OGF document process as a 'Proposed OGF Standard'.

The standard centers around a rendering neutral Core specification[1], which is currently accompanied by two renderings[4, 2]. These rendering profile the core specification with specific data query and access mechanism, and are tailored to create service which interface to XML and relational databases as backends, respectively.

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, Grid-Sphere, . . .).

This document describes a dais client package, which is supposed to provide an client side representation of the DAIS service interfaces in the SAGA framework. That document serves three purposes:

1. prove that a DAIS client rendering in SAGA is possible,

2. extend the SAGA API by dais compatible dataflow capabilities,

3. demonstrate that the DAIS intergration in SAGA will allow for the accomodation of more renderings and dais activities in the future.

2 SAGA DAIS API

2.1 Introduction

2.2 Specification

```
package saga.dais
{
  class service
  {
    ctor (url);
    dtor (void);
  }

  class basic_activity : extends saga::async::task
  {
  }

  class query : extends basic_activity
  {
  }

  class transfer : extends basic_activity
  {
  }

  class transform : extends basic_activity
  {
  }

  class deliver : extends basic_activity
  {
  }

  class dataflow : extends saga::async::task_container
}
```

Extensions to the activity types MUST live in the `saga::dais` namespace. Only a limited set of ac tivities should be part of the SAGA DAIS extension spec – however, a mechanism to create domain specific and/or implementation specific extensions should be seemless. A global repository (at `http://saga.ogsadai.org/`

or `http://ogsdai.saga.net/`) should be encouraged, for both specificastions and code.

`dataflow` adds semantics to the task container: an added activity MUST fit to the previously added activity (like port types).

2.2.1 Examples

```
─────────────────── Code Example ───────────────────
1    // C++ example
2
3    #include <saga.hpp>
4
5    int main ()
6    {
7      using namespace saga::dais sd;
8
9      sd::service   ds;
10
11     sd::query     qa = ds.create_query     ('sql://some.host.net/database',
12                                              '<sql query string>');
13     sd::transform xa = ds.create_transfer  ('other.host.net');
14     sd::transform ta = ds.create_transform ('zip');
15     sd::deliver   da = ds.create_deliver   ('to_file',
16                                              saga::url ("file:///tmp/out.dat"));
17
18     sd::dataflow  df;
19
20     df.add_activity (qa);
21     df.add_activity (xa);
22     df.add_activity (ta);
23     df.add_activity (da);
24
25     df.run ();
26
27     df.wait ();
28
29     return (0);
30   }
```

3   Conclusions

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

References

[1] M. Antonioletti, M. Atkinson, A. Krause, S. Laws, S. Malaika, N. Paton, D. Pearson, and G. Riccardi. Web Services Data Access and Integration - The Core (WS-DAI) Specification, Version 1.0. Grid Forum Document GFD.74, 2006. Open Grid Forum.

[2] M. Antonioletti, B. Collins, A. Krause, S. Laws, J. Magowan, S. Malaika, and N. Paton. Web Services Data Access and Integration - The Relational Realisation (WS-DAIR) Specification, Version 1.0. Grid Forum Document GFD.76, 2006. Open Grid Forum.

[3] T. Goodale, S. Jha, H. Kaiser, T. Kielmann, P. Kleijer, A. Merzky, J. Shalf, and C. Smith. A Simple API for Grid Applications (SAGA). Grid Forum Document GFD.xx, 2007. Open Grid Forum.

[4] S. H. M. Antonioletti, A. Krause, S. Langella, S. Lynden, S. Laws, S. Malaika, and N. Paton. Web Services Data Access and Integration - The XML Realization (WS-DAIX) Specification, Version 1.0. Grid Forum Document GFD.75, 2006. Open Grid Forum.