



NextGRID Data Processing Use Cases and Requirements

Editors:	Clive Davenhall	UEDIN

Date	Author	Comments	Version	Status
7 Aug. 2007	Clive Davenhall	Initial Version	0.1	Draft
17 Sept. 2007	Clive Davenhall	Draft for comments	0.2	Draft
9 Oct. 2007	Clive Davenhall	Modified following review	0.3	Draft
12 Dec. 2007	Clive Davenhall	Removed references to EMS standards other than BES; added mention of UDAP.	0.4	Draft
12 Feb. 2008	Clive Davenhall	Modified following review	0.5	Draft
1 Mar. 2008	Clive Davenhall	Final edits.	0.6	Draft
11 Mar 2008	D. Snelling	Final QA	1.0	Final



- 1 INTRODUCTION..... 3**
- 1.1 Data Processing Profile Overview 3
- 1.2 Relevant Specifications..... 3
- 1.3 Terminology 4
- 2 REQUIREMENTS 5**
- 2.1 Processing Data in NextGRID 5
- 2.2 Activities and Orchestrations..... 6
- 2.3 Activities 6
- 2.4 OGSA EMS 7
- 3 USE CASES..... 7**
- 3.1 Execution 7
 - 3.1.1 Direct Activity Execution 7
 - 3.1.2 Indirect Activity Execution 8
 - 3.1.3 Normal Job Termination 9
 - 3.1.4 Job Termination Invoked by the Client..... 9
- 3.2 Additional Use Cases 10
- 4 DEFINITIONS OF ACTIVITY 10**
- REFERENCES 11**



1 Introduction

This document discusses the NextGRID Data processing use cases and requirements that accompany the *NextGRID Data Processing Profile 1.0* [1].

The NextGRID Generalized Specifications aim at capturing NextGRID architectural concepts in a set of profiles that may be composed together. These profiles are specified in such a way that they could be implemented in terms of other well known specifications. While overall consistency is achieved at the conceptual level, and captured through the motivating Use Cases accompanying each Generalized Specification, system implementations, which may be based on other specifications, may not be consistent with this profile. Thus, each profile defines a realisation of the underlying concept that can be implemented. However, implementers of the NextGRID architecture may need to support multiple underlying specifications.

1.1 Data Processing Profile Overview

The NextGRID Data Processing Profile is intended for use when implementing data processing services that are in line with the concepts of NextGRID [2]. It mandates the adherence to a certain set of specifications and clarifies their use. A service implementation that uses those specifications in a manner conformant with the Profile may be said to be an “implementation of the NextGRID Data Processing Profile 1.0” or, informally, to be a “NextGRID Data Processing Service”.

Data processing is the transformation of input data (typically from one or more sources) into output data (typically destined for one or more sinks). The transformation performed is arbitrary. The implementation of such a transformation within the NextGRID architecture may be arbitrarily complex. In the simplest case the transformation may occur as a single activity indivisible within the architecture. Alternatively, a collection of such indivisible activities may be orchestrated into a workflow to perform the required transformation.

1.2 Relevant Specifications

UDAP Framework

The UDAP (Universal Dynamic Activity Package) [3] framework provides the means to realise NextGRID Data Processing Services as UDAP services. UDAP services interact with components of the NextGRID environment in order to process data.

NextGRID Workflow Language Profile

NextGRID workflows conform to the *NextGRID Workflow Language Profile* [4].



NextGRID Data Transfer Profile

Data transfer occurs according to the *NextGRID Data Transfer Basic Profile* [5].

NextGRID Naming Profile

Service naming conforms to the *NextGRID Naming Profile* [6].

OGSA-BES

Every activity performing a data transformation is supervised by a NextGRID service. This supervisory service conforms to the OGSA-BES (Basic Execution Service) specification [7].

JSDL

The data transformation that an activity is to perform is specified using the JSDL (Job Submission description Language) specification [8].

1.3 Terminology

This section introduces the terminology used in the remainder of this document. Hereafter terms defined in this section are shown in **bold** in this document. See also Section 4 and the OGSA *Glossary of Terms* [9].

Data processing: the transformation of input data (typically from one or more sources) into output data (typically destined for one or more sinks).

Data processing service: A NextGRID service which performs **data processing**. Such services are specified in the *NextGRID Data Processing Profile* [1].

Client: a NextGRID entity which wishes to invoke a **data processing service** to perform some **data processing** task.

Activity: a NextGRID service which performs a particular, defined data transformation (or instance of **data processing**; see also Section 4). An **activity** cannot be decomposed further within the NextGRID architecture. An **activity** is instantiated as a UDAP service [3] and superintended by a BES service [7].

Job: an **activity** which executes on a schedulable sub-system. **Jobs** are a subset of **activities**.

Job description: an abstract description which specifies the data transformation (or instance of **data processing**) which an **activity** or **job** is to perform. In the simplest configuration a **client** will send a **job description** to a BES service, which will instantiate an **activity** with the required description.

Job manager: a service which manages one or more **job** instances. The **job manager** encapsulates all aspects of **job** execution. It may schedule, *inter alia*, **jobs**, discharge reservations, handle failures and restart failed **jobs**.

Orchestration: a collection of **activities** (or other, subsidiary, **orchestrations**) acting in concert to perform some **data processing** task.

Workflow: a type of **orchestration** in which the constituent **activities** are marshalled according to a pre-defined pattern.

Workflow manager: an entity responsible for superintending the execution of a **workflow** to ensure that its constituent **activities** are executed according to a pre-defined pattern.

2 Requirements

2.1 Processing Data in NextGRID

Data processing is the transformation of input data (typically from one or more sources) into output data (typically destined for one or more sinks). Such transformations can be arbitrarily complex. In NextGRID the services which perform such processing are called **activities**. NextGRID **data processing** has the following characteristics.

1. An **activity** cannot be further decomposed within the context of the NextGRID architecture. Such **activities** are ‘atomic’ in the original meaning of the word. Often an **activity** will correspond to an executable binary image or interpreted script running on a single machine. However, it could also be a single portal or interface to a distributed task executing on a parallel architecture. The defining characteristic of an **activity** is that it cannot be decomposed further within the context of the NextGRID architecture.
2. Disparate **activities** can also be linked so that they act in concert to perform some more complex task. Such a set of **activities**:
 - can exchange intermediate datasets,
 - might be scheduled or staged in time,
 - might be geographically distributed,
 - could have some elements executing in parallel.
3. In NextGRID, a collection of **activities** combined to perform some **data processing** task is known as an **orchestration**.



4. A type of **orchestration** supported by NextGRID is the **workflow**. In a **workflow**, the constituent **activities** are marshalled according to a pre-defined pattern. Typically this marshalling will be performed by a **workflow manager**.
5. Any **orchestration**, howsoever complex, can be viewed 'from the outside' as a black box with a set of inputs and outputs. The **orchestration** is functionally equivalent to a single **activity** taking the same inputs and using an identical transformation to produce the same outputs.
6. Input and output datasets for NextGRID **activities** can be:
 - referred to according to the *NextGRID Naming Profile* [6],
 - input to or output from the **activity** according to the *NextGRID Data Transfer Profile* [5]. Specifically, an **activity** should be able to act as either a source or sink of data in the context of the *Data Transfer Profile*.
7. The indirection available in NextGRID Names [6] facilitates the use of temporary copies, caches *etc.*, as required within a complex **orchestration**.
8. Similarly, temporary datasets can be staged in and out of an **activity** or **orchestration**, as required.

2.2 Activities and Orchestration

Within NextGRID, the mechanisms for orchestrating complex sets of **activities** are specified in the NextGRID 'Orchestration' Generalised Specifications. Most notably, **workflows** are specified by the *NextGRID Workflow Language Profile* [4], which is accompanied by the *NextGRID Workflow Use Cases* [10]. **Workflows** are specified in terms of the NextGRID Virtual Infrastructure Model or Grid VIM [11], [12], which, in turn, is based on a Process Object Model (POM) [13] view of a **workflow**.

Conversely, NextGRID **activities** are specified in the *NextGRID Data Processing Profile* [1]. Section 3 (below) presents use cases for **activities** which conform to the *NextGRID Data Processing Profile*.

The NextGRID Orchestration Profiles and the *NextGRID Data Processing Profile* are complementary.

2.3 Activities

There are various ways that an **activity** might be instantiated: an executable running on a single processor, a series of executables running on a cluster of parallel machines, an interface to a job scheduler providing delayed execution *etc.* However, they all have the characteristic that they are indivisible within the context of the NextGRID architecture, in all cases being viewed as an atomic task.

The sequence of events to instantiate an **activity** is as follows.

1. A BES service [7] is invoked with a JSDL [8] **job description** prescribing the data transformation (that is, the **data processing** task) to be performed.
2. The BES service instantiates a UDAP service [3] as the data processing service.
3. The UDAP service performs the **activity** and handles any necessary communication with other NextGRID services.

2.4 OGSA EMS

The OGSA (Open Grid Services Architecture) EMS (Execution Management Services) [14] is an architecture for initiating, managing and controlling **activities** executing in a Grid environment (for example, see *The Open Grid Services Architecture* [15], Section 5.4, p20). Thus, in the context of this document, EMS is an architecture for executing **activities**.

EMS is an architecture and not a standard (and thus forms part of the OGSA ‘informational’ rather than ‘recommendation’ track of documents and specifications). However, OGSA uses EMS to identify and position standards which address parts of the ‘activity management’ problem. This effort continues, and all the necessary standards are not yet in place. The following section presents use cases for **activity** execution using BES taken from the *OGSA EMS Architecture Scenarios* [14].

3 Use Cases

This section presents some basic use cases derived from the fundamental operations that can be performed on an **activity**.

3.1 Execution

Execution involves running an **activity** to perform some **data processing** task. The four scenarios below are taken from the *OGSA EMS Architecture Scenarios* [14], Section 2, pp4-9. The terminology has been revised to conform that used in the present document.

3.1.1 Direct Activity Execution

In this context ‘direct **activity** execution’ means that the **client** directly invokes a BES service. This BES service instantiates the **activity**.

Assumptions:

- The **client** knows which BES service to use.
- The BES service is already set up.

Entities:

- The **client**.
- The remote BES service to be invoked.

Description:

Direct **activity** execution proceeds as follows.

1. The **client** contacts the BES service directly and submits a request for the BES service to perform the **activity**. The **client**'s request will include a **job description**.
2. The BES service instantiates the **activity** as a UDAP service. The UDAP service performs the **activity**. Typical facilities provided by the UDAP service include: resource discovery, matching, claiming, and accounting.
3. The **client** monitors progress directly by interacting with either the BES or UDAP service.

3.1.2 Indirect Activity Execution

In this context 'indirect **activity** execution' means that the **client** delegates the invocation of a BES service to a **job manager**, rather than invoking it itself. This method of working permits additional flexibility. The BES service instantiates the **activity**.

Assumptions:

- The **client** knows which **job manager** to use.
- The **job manager** knows which BES service to use.
- The BES service is already set up.

Entities:

- The **client**.
- The **job manager**.
- The remote BES service to be invoked.

Description:

Indirect **activity** execution proceeds as follows.

1. The **client** contacts the **job manager** and provides a **job description** specifying the **activity** to be performed.
2. The **job manager** submits a request for the BES service to undertake the **activity**. Often the **job manager** will schedule the **activity** so that it runs as a **job**, but this is not required.
3. The BES service instantiates the **activity** (or **job**) as a UDAP service. The UDAP service performs the **activity**.

4. The **job manager** monitors progress directly by interacting with either the BES or UDAP service.

3.1.3 Normal Job Termination

In normal job termination, the **client** simply waits for the executing **activity** to complete (contrast with termination invoked by the **client**, below).

Assumptions:

- The **client** knows how to contact the BES service managing the **activity**.

Entities:

- The **client**.
- The BES service.
- The **activity**.

Description:

The **client** simply waits for the **activity** to terminate normally. The **client** can detect that the activity has completed in two ways: (i) by polling the BES service or (ii) by the BES service sending a notification to the **client** when the **activity** has terminated.

3.1.4 Job Termination Invoked by the Client

Here the **client** terminates the **activity** while it is still executing.

Assumptions:

- The **client** knows how to contact the BES service managing the **activity**.

Entities:

- The **client**.
- The BES service.
- The **activity**.

Description:

The **client** instructs the BES service running the **activity** to terminate the **activity**. The BES service duly terminates the **activity** and releases any resources that it was using.



3.2 Additional Use Cases

The *OGSA EMS Architecture Scenarios* [14] include a number of additional use cases, which are equally applicable to NextGRID **activities**. Specifically it presents use cases in the following areas:

- Selected **job** (or **activity**) execution, in which the **job manager** chooses a BES service rather than being pre-configured with the one that it is to use (Section 3, pp9-10).
- Deploying and un-deploying the application which the **activity** runs in order to perform the required instance of **data processing**, rather than having a pre-deployed or hard-wired application (Section 4, pp10-17).

4 Definitions of Activity

This document uses the definition of **activity** given in Section 1.3: ‘a NextGRID service which performs a particular, defined data transformation (or instance of **data processing**).’ This definition is consistent with the one used in the UDAP framework and various OGSA documents. This section demonstrates this consistency by quoting the definitions given in relevant documents.

- *UDAP Use Cases and Requirements* [16], Section 2.1, ‘Definition of Activity’, first two paragraphs:

‘An activity is a *unit of work* on a Grid. It can be a job; a task; a data processing operation; a data access operation; an application execution; a program execution; a Web Service invocation; something that a user or application needs to do, take care of, or execute.

‘An activity is atomic. This means that an activity is an indivisible unit of work from an activity management perspective. When you stop an activity, you stop all of it, not any one part of it.’

- GFD-I.106: *OGSA EMS Architecture Scenarios* [14], Section 6: ‘Glossary’, p18:

‘Activity: The smallest part of possibly a larger sequence of activities generated from a single submitted job.

‘In this document a job consists of exactly one activity and the terms “job” and “activity” are sometimes used interchangeably.’

- GFD-R-P.108: *OGSA Basic Execution Service* [7], Section 1, ‘Introduction’, p4:

‘This Basic Execution Service (BES) specification defines Web Services Interfaces for creating, monitoring and controlling computational entities such as UNIX or Windows processes, Web Services, or parallel programs – what we call *activities* – within a defined environment. Clients define activities using the Job Submission Description Language (JSDL).’

The way that ‘activity’ is being used in the above documents clearly derives from one of its dictionary definitions as: ‘an organizational unit for performing a specific function’ [17].

References

- [1] C. Davenhall, *NextGRID Data Processing Profile*, V1.0, NextGRID Project, 11 March 2008.
- [2] D. Snelling, M. Fisher, A. Basermann, F. Wray, P. Wieder and M. Surrudge, *NextGRID White Paper*, v0.5, NextGRID Project, July 2005, http://www.nextgrid.org/download/publications/NextGRID_Architecture_White_Paper.pdf
- [3] A. Anjomshoaa and P. Wieder, *UDAP Schema*, v1.0, NextGRID Project, February 2008.
- [4] J. Ferris, B. Cantalupo and N. Matskanis, *NextGRID Workflow Language Profile*, V1.0, NextGRID Project, March 2008.
- [5] M. Drescher, S. Davey and A. Anjomshoaa, *NextGRID Data Transfer Profile*, v1.0, NextGRID Project, 19 February 2008.
- [6] S. Davey and D. Snelling, *NextGRID Naming Profile*, V1.0, NextGRID Project, 26 February 2008.
- [7] I. Foster, A. Grimshaw, P. Lane, W. Lee, M. Morgan, S. Newhouse, S. Pickles, D. Pulsipher, C. Smith and M. Theimer, GFD-R-P.108: *OGSA® Basic Execution Service*, V1.0, Open Grid Forum, 8 August 2007.
- [8] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher and A. Savva, GFD.56: *Job Submission Description Language (JSDL)*, V1.0, Open Grid Forum, 7 November 2005.
- [9] J. Treadwell, GFD-I.120: *Open Grid Services Architecture Glossary of Terms*, V1.6, Open Grid Forum, 12 December 2007.
- [10] J. Ferris and N. Matskanis, *NextGRID Workflow Use Cases*, V1.0, NextGRID Project, March 2007.
- [11] M. Surrudge (ed), D5.5: *NextGRID Workpackage Progress Report: Grid Dynamics*, NextGRID Project Deliverable, 02 April 2007 (Grid VIM).
- [12] N. Matskanis and B. Cantalupo, P5.3.4: *Updated Adaptive Workflow Enactor and OWL-WS Parser Implementation*, NextGRID Project Output, 29 September 2006 (Grid VIM).
- [13] R. Ruiz, P6.5.2: *Adaptive API Design and Reference Implementation*, NextGRID Project Output, 19 April 2007 (POM).
- [14] A. Savva, H. Kishimoto, S. Newhouse and D. Pulsipher, GFD-I.106: *OGSA® EMS Architecture Scenarios*, V1.0, Open Grid Forum, 18 April 2007.
- [15] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaian, J. Treadwell, J. Von Reich, GDF-I.80: *The Open Grid Services Architecture*, V1.5, Open Grid Forum, 24 July 2006.



- [16] A. Anjomshoaa, H. Mersch and Ph. Wieder, *UDAP Use Cases and Requirements*, V1.0, NextGRID Project, March 2008.
- [17] Merriam-Webster On-line Dictionary: <http://www.m-w.com>