Comparison of European Grid Projects

Jarek Nabrzyski, Ariel Oleksiak (PSNC)

Project:

GEMSS

Area:

Application Environment and Tools

Table of Contents

1. Introduction

1.1. Objective and Structure

This document is one of thirteen templates that have common goal to gather information related to main European Grid Projects in order to make their accurate comparison in the framework of GRIDSTART initiative. We believe that the participation of particular projects members in preparation of this document will allow comparing all activities in a credible and exhaustive way.

The proposed structure of the description consists of two parts. The former is concerned with the general overview and architecture together with the contents of layers (the first template). The latter includes the main components of the Grid infrastructure (remaining 12 templates). Since information regarding the project architecture is to be quite general, more detailed description should be provided in the review of the main aspects of the Grid infrastructure. In order to prepare uniform description for each project, we identify the important issues that have to, should or can be included into particular components. Common issues for all components and these specific for this component are briefly described in the next section.

We ask you to proceed according to this schema. However, a feedback is obviously welcome. For some projects the document has been partially completed on the basis of descriptions found at the official web pages. In this case, we ask you to revise already filled in sections, correct and complete them if necessary.

You should take into consideration future plans while you fill in particular sections. Actually they are even more important then the current state of the project components. If you are not going to design some elements in the scope of the project at all, please, note it in the proper section.

1.2. Uniform description

All the descriptions of the Grid infrastructure components are divided into three parts: **General** section includes main requirements and functionality, **Details** section relates to the issues specific for particular component and **External** defines its connections with other components and users.

As it was mentioned above, some of the issues are common for all components or at least repeat for many of them. Such issues, appearing for many or even all areas are shortly characterized below.

In **General** section:

Main requirements determine the objectives and requirements of the workpackage or the software module responsible for the design of functionality related to the particular domain of the Grid infrastructure.

Functionality contains a set of operations provided by the project in the given area.

In **External** section:

Interfaces define services, SDKs, APIs and so forth which can be used in order to access the functionality of the component.

Low level Grid middleware is the middleware providing basic Grid functionality as for example Globus or UNICORE.

Relations with other components determine components that utilize or are utilized by component being described as well as data and information flow between them.

Issues that are specific for this particular domain of the Grid infrastructure are presented in the sequel. Some of them, which we consider to be clear, have been skipped, however, if they turn out to be vague, please, contact the authors of this document (ariel@man.poznan.pl).

The **Details** section contains description of tools and environment that together support developers during the process of the Grid-enabled application design as well as facilitate using this application in Computational Grids.

Support for adaptability specifies whether the project provides methods (e.g. API with SDK) for program adaptation to dynamic changes in the Grid. The application check-pointing feature can be the example of such method.

2. Application Environment and Tools

2.1. General

GEMSS includes 6 medical prototype applications which provide image processing and simulation services via the GEMSS test-bed. All components have to be Grid-enabled and interfaced. This document presents a snapshot of current developments. The GEMSS design is still being discussed by the partners and not all issues have been finalized yet.

- **Main requirements**
  GEMSS defines 6 different medical application services, with very heterogeneous requirements. The applications are based on pre-existing components (3D visualization, medical image processing, mesh generation, Finite Element simulation or computational fluid dynamics etc.) to which we do not have source access in all cases. We want to minimize the intervention needed to insert applications into Grid services and to use them over the Grid. Most of the services need HPC facilities (like PC clusters) on the server side. Some applications also require compute resources on the client side (3D visualization, interaction with the model, manipulation of input / intermediate / output results).

- **Functionality**
  The main focus is on remote batch applications. This may be sufficient for GEMSS on the server side. The applications are planned to provide a load vector that allows the determination of the resources that are necessary to run the job on the Grid.

2.2. Details

- **Supported application types**
  Simulation applications with response times ranging from near-real-time to several days or weeks.
  Some applications may be interactive in the sense that a user needs to interactively manipulate input, intermediate results or output.

- **Support for adaptability**
  Planned is flexible scheduling of jobs with possibility to reserve slots for “interactive” session in advance.

- **Tools**

  Debugging
  Current plan is that each medical application service will provide a test-suite of benchmark examples with reference results that can be used to validate proper functioning.

  Application performance analysis
  GEMSS does not seek to monitor or manage computers directly. GEMSS is designed to work with local cluster management technology. It will create a model of the way the cluster operates and use experience from previous runs to predict behaviour and availability of services in order to negotiate about expected QoS.

  Executables and source code management
  Applications are not developed in a common source code tree.

  Other

2.3.External

- **Interfaces**

  Overview
  This is application specific, while some codes have to be integrated “as is”, we have source access to others. Where possible and appropriate we plan to have interfaces for monitoring application specific behaviour (i.e. convergence of a solver) and application check-pointing (if supported by the application or operating system).

  Programming language support
  Applications may be written in any programming language. Source code may not be available.

  Platform support
  Most applications run on Intel/Linux clusters.

  Software methodology (e.g., object oriented dynamically linked library)
  Software methodology is application specific, i.e. dependent on the (proprietary) codes.

- **Low level grid middleware**

GEMSS will build on top of existing Grid and Web technologies, maintaining compliance with standards (OGSA). We will use our own web service based developments and explore the use of UNICORE and Globus GT3.

- **Utilization of software components**
  The GEMSS application software belongs to the partners. The project is not at liberty to reveal details.