

Earth science test suites to evaluate grid tools and middleware—examples for grid data access tools

W. J. Som de Cerff · M. Petitdidier · A. Gemünd ·
L. Horstink · H. Schwichtenberg

Received: 17 September 2008 / Accepted: 26 February 2009 / Published online: 31 March 2009
© Springer-Verlag 2009

Abstract The Earth Science (ES) community has three major IT related concerns: Modeling (computing intensive), exploitation of datasets, and production of large shared datasets. All could be accomplishable using Grid. Different Grid middleware solutions exist and are being developed. In DEGREE (Dissemination and Exploitation of Grids in Earth sciencE) the aim was to disseminate and promote uptake of Grid in ES and to create a bridge between ES and Grid communities. DEGREE identified key ES requirements and has disseminated these to Grid projects, evaluated Grid middleware tools and standards regarding ES requirements and provided feedback to Grid developers. In order to convey requirements to the Grid community, test suite specifications were developed. Test suites provide real applications for testing functional and non-functional aspects of Grid, contrary to typical whiteboard tests or use cases. In this paper the GOMEVAL test suite is explained in detail and the results obtained are discussed.

Keywords Earth science · Grid · Requirements · Test · Test suites

Communicated by H.A. Babaie

W. J. Som de Cerff (✉) · L. Horstink
Royal Netherlands Meteorological Institute, KNMI,
P.O. Box 201, NL-3730 AE De Bilt, The Netherlands
e-mail: sdecerff@knmi.nl

M. Petitdidier
Laboratoire Atmosphères, Milieux, Observations Spatiales,
10-12 Avenue de l'Europe,
78140 Velizy, France

A. Gemünd · H. Schwichtenberg
Fraunhofer Institute for Algorithms and Scientific Computing
(SCAI),
53754 Sankt Augustin, Germany

Introduction

Earth Science (ES) is an all-embracing term for sciences related to the planet earth, covering various areas like atmosphere, ocean, solid earth and their interfaces. ES has a large and diverse user community with universities, research institutes, industries and governmental organizations. Characteristics of this scientific domain are that any observation is unique as it cannot be repeated with the same environmental conditions, and that it depends on four coordinates, three spatial dimensions and time. In addition to the research on the solar-earth system, the ES community has to inform on crucial societal issues like probability of natural disasters, long-term climate change and energy reserves.

For all those tasks, the main needs of the ES community may be summarized as follows. The increased complexity of ES applications leads to more intensive computing and access to many spatially-distributed data archives, databases and physical models. The existing computing resources and the access to distributed data at the data centers available in one institute or organization are in general not sufficient. Often datasets are too large to be copied by each end-user or even stored in a single location. Due to data policy restrictions some datasets cannot be copied and distributed in different locations. In those cases, the data provider should also host computing resources able to process the data or provide secure access to distributed computing resources. Another important need is the ability to use specific ES environments on different available computing systems; for this goal many web services have been developed. The experience with web services and metadata handling is growing rapidly in the ES community. Developments and techniques in the ES community include web services (including Open GIS), Open Source software,

interoperability of data and metadata. New technology, such as Grid, has to take into account the ES needs and knowledge base.

The ES community has investigated Grid technology since 2000 through different Grid projects, like DataGrid and EGEE (Enabling Grids for E-science; <http://www.eu-egee.org>). Despite the enthusiasm of some ES pioneers, it is a nontrivial task to establish awareness, gain new users and teach committed users to effectively use the Grid infrastructure. Because of the complex applications used in Earth Science, there persists a significant gap between the ES domain and Grid application domain. The experience acquired around the world with several academic and industrial applications has demonstrated that Grid infrastructures could respond to the complexity and constraints imposed by Earth Science applications. However the interface between ES environment and Grid middleware is not straightforward for many applications.

In order to facilitate the adoption of Grid by the ES community, an EU Specific Support Action was set up, the DEGREE project (Dissemination and Exploitation of Grids in Earth science). DEGREE had to bridge the gap between the ES and Grid communities and to make the Grid developers familiar with the ES needs. DEGREE examined both the ES usage and the Grid tools available for data management, job control, job management, and portals. The list of ES requirements initiated by the DataGrid project was expanded to include new emerging requirements (Som de Cerff et al. 2007a, 2008). Some general ES requirements on the reliability, quality of service, stability and security of the infrastructure are valid in all scientific and technical domains; other requirements are more ES specific, especially in the area of data management.

However, providing only requirements to Grid middleware developers is not sufficient, as they might misunderstand or misinterpret requirements by lack of domain knowledge. The common way to check if requirements are fulfilled is to build generic use cases relevant to one or a few requirements and test them on the middleware or tools. Use cases are good to test functionalities, but are of limited use to test scalability or application workflows. Therefore the methodology adopted was to collect a large set of typical application scenarios or a family of applications and to analyze them in terms of requirements. The status of each requirement can be tested with an actual application example. From this collection of applications a set of ES applications was selected that cover all main ES requirements. From this set, test suites were created.

The second part of this paper describes the methodology used and subsequently the test suites created, their definition, and their characteristics. In the third part, test suite setup and usage is demonstrated with the ES-TS-GOMEVAL test suite, used for the validation of satellite

ozone profiles. This test suite was used to evaluate different Grid access tools.

Test suites

Methodology

A list of ES requirements was created in 2000 during the first EU Grid project, DataGrid. This list was updated during the different EGEE projects (EGEE and EGEEII; Som de Cerff et al. 2007a). However, with time going, gaps are still present between, on one side, the ES needs and, on the other side, the Grid middleware and tools available. The role of the DEGREE project was to evaluate the gaps and to survey their status as technology is evolving fast. To carry out this work a methodology was needed and defined.

In software development, structured methods exist for testing requirements (Hambling et al. 2006). But these methods are aimed at projects, where the sponsor of the project states the requirements. In this case there is a clear relation between the definer of the requirements and the software developers. The ES case is different. ES is neither sponsor of Grid development projects nor sponsor of Grid infrastructure projects, therefore a different approach for requirement verification is needed.

Grid middleware and tools are in general evaluated through the fulfillment of a list of requirements provided by the application domain and/or use cases. The list of ES requirements prior to the DEGREE project did not address Grid functionalities directly, because of the strong reference to the EGEE projects. In addition to this, the use cases needed to test requirements had to be simple. They were designed to test only a few functional requirements of the middleware or tool at a time. They did not test the non-functional requirements, like scalability. Example: several Grid access mechanisms to databases exist, as demonstrated in chapter three of this document. Their performance and usability depends on the number of records and/or the number of accesses in input and output. Therefore the performance can only be tested with complex use cases based on real ES applications.

Because of this it was decided to build what we call a ‘test suite’. Based on actual applications a test suite consists of a set of test cases, where each test case addresses specific ES requirements. The test suite specifications aim to provide high value input from the ES applications community towards the Grid middleware development community. The final aim of the generated inputs is to evaluate and confirm the extent to which the Grid middleware are capable of handling specific ES requirements.

The test suites were presented to Grid infrastructure and tool developers for tests and/or feedback. The results point

out that this approach is very fruitful for collaboration between the ES and Grid communities (Som de Cerff et al. 2008).

Definition

Figure 1 gives an overview of the different components of a test suite (requirements, test cases, and test procedures) and their interdependencies. For the test suites the following definitions are used:

- Application: An Earth Science application. This can consist of multiple programs (application code or binaries) and data sets, aiming at answering an Earth Science scientific question or aiming at producing products for further research
- Test Suite: Describes how to use an Earth Science application to test certain parts of Grid middleware. The application will be decomposed into test cases and therefore a Test Suite consists of a series of Test Cases
- Test Case: A test case is a part of an Earth Science application to test a certain part of Grid Middleware. A test case will cover one or more requirements and will be composed of one or more Test Procedures.
- Test Procedure: Belongs to a Test Case. A Test Procedure describes the test to be carried out in a detailed way, including pass and fail criteria.

In this way requirements can be matched to test cases and test procedures. The coverage of the requirements with the test cases is specified for each test suite. A test case can be tested without running the whole test suite.

A template was defined and used to describe the individual test suites. The test suites definition is documented in the test suite specification document (Som de Cerff et al. 2007a, b), which also provides a high level

overview of all available test suites and a matrix matching requirements and test suites. The individual test suite description documents are gathered in two documents, the first aims at data management, the second aims at job management and job control (Schwichtenberg et al. 2007; Tran et al. 2007). Of course individual test suites, together with their data and application are also available.

ES test suites

Test suite selection

Results from previous Grid projects are used for the definition of ES requirements. Around 20 ES applications were collected and classified into application families by complexity. Complexity is defined by the requirements (Som de Cerff et al. 2007a, b). Test suites were built with a selection from these applications. Not all the collected applications can be provided as a test suite, because some consist only of a single algorithm and for others the software and/or data are not available due to license restrictions. It is fundamental to the test suites that the software and data are freely available for any Grid developer.

Table 1 provides the list of the test suites available for Grid developers. The table describes the test suite reference name, a short application description, to which application family it belongs and its special focus on specific Grid components. To capture the evolving ES applications the applications are grouped into families of applications rather than application domains. This eases the maintenance (e.g. changing priorities, adding new requirements) and tracing of requirements. Three families have been identified, following a scheme with three different levels of complexity. In this case complexity refers to the requirements placed on

Fig. 1 Schema: Test suites, requirements, test cases and test procedure

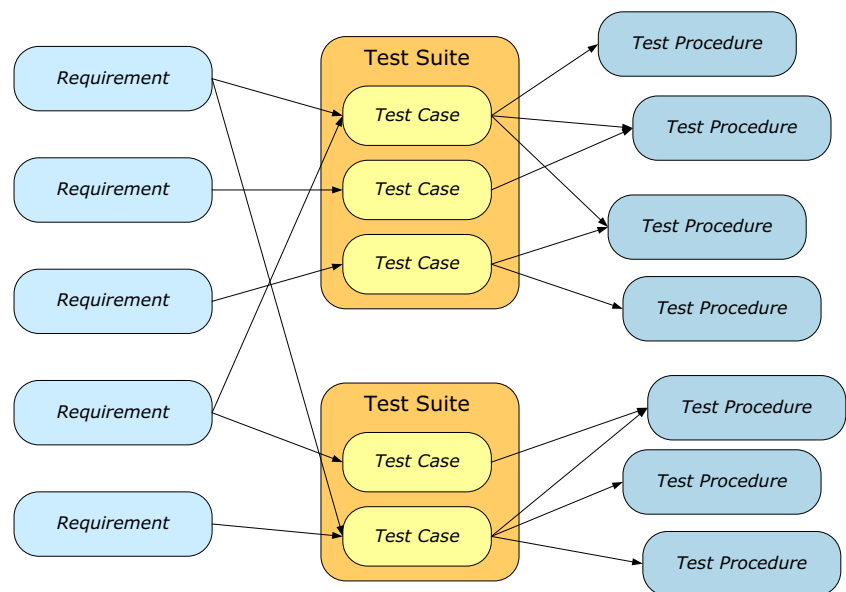


Table 1 Selected applications used to create test suites

Test suite reference	Application	Family	Special focus
ES-TS-GOMEPRO	GOME level 2 product processing	Simple	Data
ES-TS-CMT	Earthquake: Centroid Seismic Moment Tensor	Simple	Job control & management
ES-TS-GRIMI	GRIMI 2 test suite	Complex	Data, Portal
ES-TS-GOMEVAL	GOME validation	Complex workflow	Data, Portal Job control & management
ES-TS-SPIDR	Space Physics Interactive Data resources (SPIDR)	Complex workflow	Data, Portal Job control & management
ES-TS-FFSC	Flood Forecasting Simulation Cascade (FFSC)	Complex workflow	Data, Portal Job control & management
ES-TS-PUMA	The Portable University Model of the Atmosphere (PUMA)	Complex workflow	Data, Portal Job control & management
ES-TS-ESSE	Environmental Scenario Search Engine	Complex workflow	Data, Portal control & management

Grid by the application, not to the application itself. The first family is called the simple applications, the second the complex applications and the third family is the complex workflow applications (Som de Cerff et al. 2007a, b).

Appendix 1 gives a list of requirements that could be tested with the test suites. In the following paragraphs the applications used to create these test suites are described with their relevant requirements.

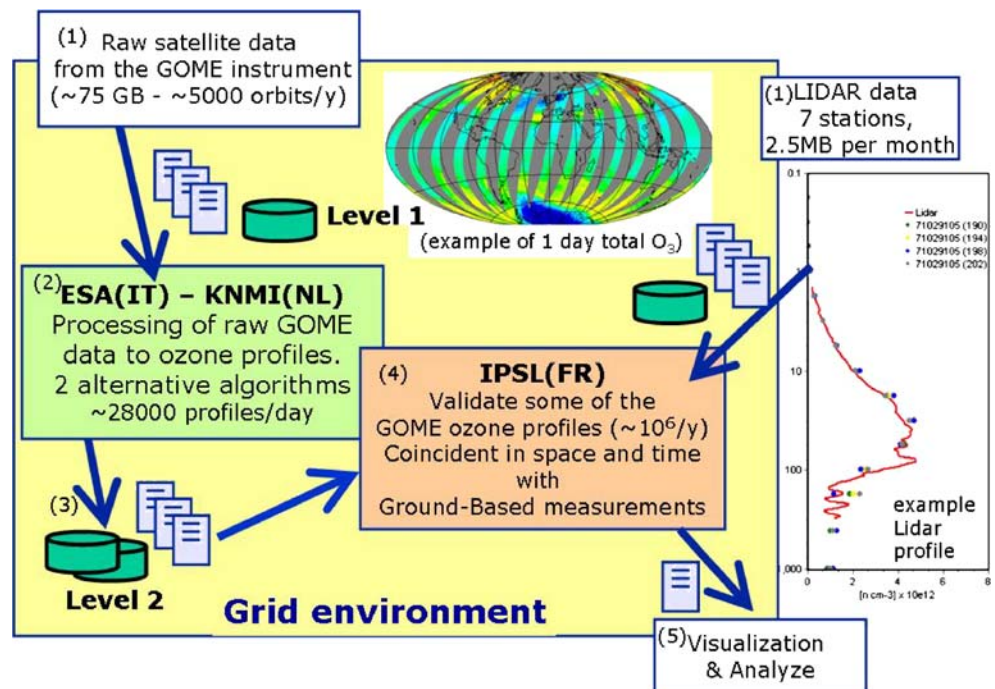
ES-TS-GOMEPRO and ES-TS-GOMEVAL test suites

The ERS2-satellite based Global Ozone Monitoring Experiment (GOME) provides UV-VIS-spectra of nadir scattered sunlight. Several products can be derived from these measurements, e.g. Ozone columns, Ozone profiles, Cloud maps and NO₂ maps.

Figure 2 presents a schematic workflow of the application relative to the test suites ES-TS-GOMEPRO and ES-TS-GOMEVAL. (1) First the raw GOME data (Level 1) needs to be uploaded (or made accessible) to the Grid, the Lidar files also need to be uploaded to the Grid. (2) The raw GOME data can then be processed using an inversion algorithm or an artificial neural network algorithm to retrieve the ozone profiles (Level 2). (3) Those profiles are stored on Grid storage elements (4) and validated using ground-based Lidar data. Results can be visualized and send to the user (5).

The *ES-TS-GOMEPRO* test suite applies an Ozone Profile retrieval algorithm to GOME data (van der A et al. 2002). This algorithm processes not only the raw GOME data but also uses files containing climatological ozone data and meteorological model outputs relative to the dates processed. The resulting Ozone profiles (level 2 products)

Fig. 2 Workflow for GOME processing (ES-TS-GOMEPRO) and validation (ES-TS-GOMEVAL) test suites



are used for climate and air quality research and ozone concentration monitoring. This test suite is a typical example of ES satellite data processing. The application can produce a large amount of (small) output files, up to several millions when processing a year of data, which makes it a good case study for regression, performance and stability testing of middleware. The major requirements of this application concern the data management, the input and output file handling and the update of the metadata catalogue as soon as new ozone profiles are processed.

The *ES-TS-GOMEVAL* test suite validates 7 years of satellite ozone profiles using Lidar observations (Iapaolo et al. 2007; del Frate et al. 2005). GOME ozone profiles are obtained using an artificial neural network algorithm on the Grid. The validation is done by comparing the ozone profiles to the ones observed by Lidar. The Lidar files are extracted from the Network for the Detection of Stratospheric Change (<http://www.ndsc.ncep.noaa.gov/>) database and need to be transferred to a storage element. The validation procedure needs to find satellite orbits collocated with a Lidar station on a given date. A metadata-base needs to be created to search for orbits passing over a Lidar site. A workflow is needed to validate simultaneously the 7-year of ozone profiles retrieved with two versions of the neural network algorithm by using the Lidar ozone profiles obtained at six different sites. The major requirements of *ES-TS-GOMEVAL* concern data management (handling of files, geospatial searches in the metadata base) and job management.

ES-TS-CMT (Earthquake: Centroid Seismic Moment Tensor) test suite

When an earthquake occurs it is important to determine as quickly as possible the main features to characterize its sources, i.e. its centroid and seismic moment tensor (Dziewonski et al. 1981). A 3D parameter space of discrete points is determined to cover the area around the earthquake location. The estimated location is published by the US Geological survey earthquake (USGS) right after sensing the event. The seismic data used in the application come from the GEOSCOPE (<http://geoscope.ipgp.jussieu.fr/>) seismometer network. Both, USGS and GEOSCOPE, are needed before running the job. The *ES-TS-CMT* application is based on an inversion algorithm. In each point of the parameter space a Green function is computed for different times until the results fit the data. It is a multi-job-parallel application, 150–300 independent jobs are submitted simultaneously, their number depending on the number of seismic stations used and on the number of parameters. The number of CPUs required depends on the number of jobs. A run normally takes 4–6 h. The volume of the input data is around 16 MB and between 30–300 MB for the output (Clévéde et al. 2009).

The major requirements of the *ES-TS-CMT* concern job control and management. First of all, it is a job on alert that must run as soon as possible after an earthquake takes place: it needs near real time execution. This requirements makes the test suite suitable for performance testing of middleware. Secondly, monitoring of the jobs is needed to check that they are all executed.

ES-TS-GRIMI-2 test suite

GRIMI-2 is the scientific prototype of the operational Envisat/MIPAS level-2 processor (Fusco et al. 2007). It makes validation with the operational MIPAS level-2 processor possible and serves as well as a reference processor, if the operational processing fails. The Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) is an instrument mounted on the ESA Envisat satellite. MIPAS is a Fourier transform spectrometer for the detection of limb emission spectra in the middle and upper atmosphere. It observes a wide spectral interval throughout the mid infrared with high spectral resolution. MIPAS detects and spectrally resolves a large number of emission features of atmospheric minor constituents playing a major role in atmospheric chemistry. Due to its spectral resolution capabilities and low-noise performance, the detected features can be spectroscopically identified and used as input to suitable algorithms for extracting atmospheric concentration profiles of a number of target species.

Processing takes approximately 10 min/file, input data size approximately 250 MB and the output data size approximately 20 MB. For each selected input file, two output files are produced. Each selected input file, requires one selected and six stable auxiliary files. The application runs best using a workflow manager.

The test shall demonstrate the capability of Grids to deal with operational processors for relatively large data volumes from satellite instrument measurements. The test-suite shall show the usefulness of Grids for the processing of real satellite data in a reduced amount of time compared to the existing non-Grid ground segments.

ES-TS-SPIDR (Space Physics Interactive Data Resource) test suite

SPIDR (Space Physics Interactive Data Resource) is a de facto standard data source for solar-terrestrial physics, functioning within the framework of the International Council for Science (ICSU, <http://www.icsu.org/>) World Data Centers (Zhizhin et al. 2008). It is a distributed database and application server network, built to select, visualize and model historical space weather data distributed across the Internet. SPIDR can work as a fully-functional

web-application (portal) or as a Grid of web-services, providing functions for other applications to access its data holdings. The main requirements are on data management, with focus on database access through web services.

ES-TS-FFSC (Flood Forecasting Simulation Cascade) test suite

The Flood Forecasting Simulation Cascade (FFSC) is a hydro-meteorological application, which tries to predict oncoming floods based on series of simulations of meteorological, hydrological and hydraulic conditions in the target area (Hluchý et al. 2005; Babik et al. 2007). The application consists of several simulation steps. Different models can be chosen for each step. The application includes pre-processing, post-processing and visualization. In case of a flood event, the application needs to be run at a high time-rate (real time). This makes the test suite suitable for performance testing. Other requirements are related to data management. The input consists of a diverse amount of meteorological and hydrological observations as well as meteorological and hydrological model output. The application requires the possibility to construct complex workflows.

ES-TS-PUMA (The Portable University Model of the Atmosphere) test suite

The Portable University Model of the Atmosphere (PUMA) is a model for atmospheric circulation developed in Fortran-90 at the Meteorological Institute of the University of Hamburg, Germany (Liakka 2006). It serves as a training tool for junior scientists, allowing them to work with a program that is easier to understand than the comprehensive general circulation model of the atmosphere ECHAM (European Centre Hamburg Model (global climate model)).

The application submits MPI jobs. As individual PUMA tasks pass around a substantial amount of data, it also provides a basic stability and performance test for computing and storage resources on Grid infrastructures.

ES-TS-ESSE Environmental Scenario Search Engine test suite

This is a typical data mining application test suite. The Environmental Scenario Search Engine (ESSE) searches for severe magnetic storms inside the Space Physics Interactive Data Resource (SPIDR). It is deployed as a standard OGSA-DAI Grid data service (Zhizhin et al. 2007). The main ESSE requirements cover data and metadata management, workflow management and system integration.

Evaluation of the ES-TS-GOMEVAL—test suite usage example

The ES-TS-GOMEVAL test suite

We have chosen a formal way to describe test results or the evaluation of the test suites in grid environments. In the previous chapters the GOME processing and validation test suites were introduced. Here a more detailed description is given of the ES-TS-GOMEVAL test suite and its use. It provides an example to illustrate the test suite setup and the tests on ES requirements. All other defined test suites follow the same structured approach. The ES-TS-GOMEVAL test suite was used for testing database interfaces in EGEE. Results of the test suite are described later on in this document. The actual test cases are not included here. They can be found in Schwichtenberg et al. (2007). The documentation of the test suite, the tools needed to extract the metadata, and the application software can be found on the DEGREE website (<http://www.eu-degree.eu>). The GOME level-2 data files, the tools needed to extract the metadata from GOME and the Lidar files can be retrieved by anonymous ftp from ESA and ND SC.

Data are at the centre of the application scenario on which this test suite is based. Therefore the main focus of the tests is on the data management components. The scenario can logically be separated into two steps. Firstly is the preparation of the metadata database from the raw data files, and secondly the selection of files for validation purpose through metadata queries.

For the first part Grid jobs running on the executing worker nodes have to retrieve the data files, which involve replica lookup and retrieval from storage elements (of course, the data has to be put on the Grid first). The data were collected from different data sources and are thus only available in different file formats. Because the scenario assumes no existing meta information, the data source and format has to be identified by the file content to apply the right metadata parser. Generated metadata are converted to the same data types and subsequently the metadata are written to a remote database, accessed through one of three different Grid database components under test.

The second part, the actual GOME validation in the sense of the scenario, consists of a Grid job, which could be called parametric, because it takes the identifier of a data file from one of the sources and finds corresponding files of the other data source by querying the metadata database.

The remarks above make clear that the most delicate part of the process is reading and writing a metadata database reliably and securely from the distributed environment. The test suite was used in the EGEE project (Enabling Grids for E-science; <http://www.eu-egee.org>) to examine three dif-

ferent solutions for this second step to compare their approaches and technicality on different software components. Enabling Grids for E-science (EGEE) is the largest multi-disciplinary grid infrastructure in the world, which brings together more than 140 institutions to produce a reliable and scalable computing resource available to the European and global research community. To make the evaluation of three different solutions possible the infrastructure of the EGEE project is complemented by additional software components. The following section introduces the components and their modus operandi and describes the tests.

Target middleware requirements

The targeted middleware requirements of the ES-TS-GOMEVAL are in descending order of importance

- Access to databases; the middleware needs to support a technique to access databases outside the Grid.
- Data management; the middleware data management facility needs to be able to handle large numbers of files with fine grained access policies and a sophisticated replication service. This replication service should be able to store metadata with spatial-temporal concepts and support the same fine grained access policies that the data management facilities support.

- Job and Workflow control; the middleware job management system needs to be able to execute workflows, like the validation workflow shown in Figure 3. Automated data and metadata discovery services are desirable for this sophisticated technique which supports semantic descriptions.
- Security of data and database access; the access to data and databases needs to be secure, because not all data is available to the public (base data (GOME, Lidar) is freely accessible, but the result data has restricted access)

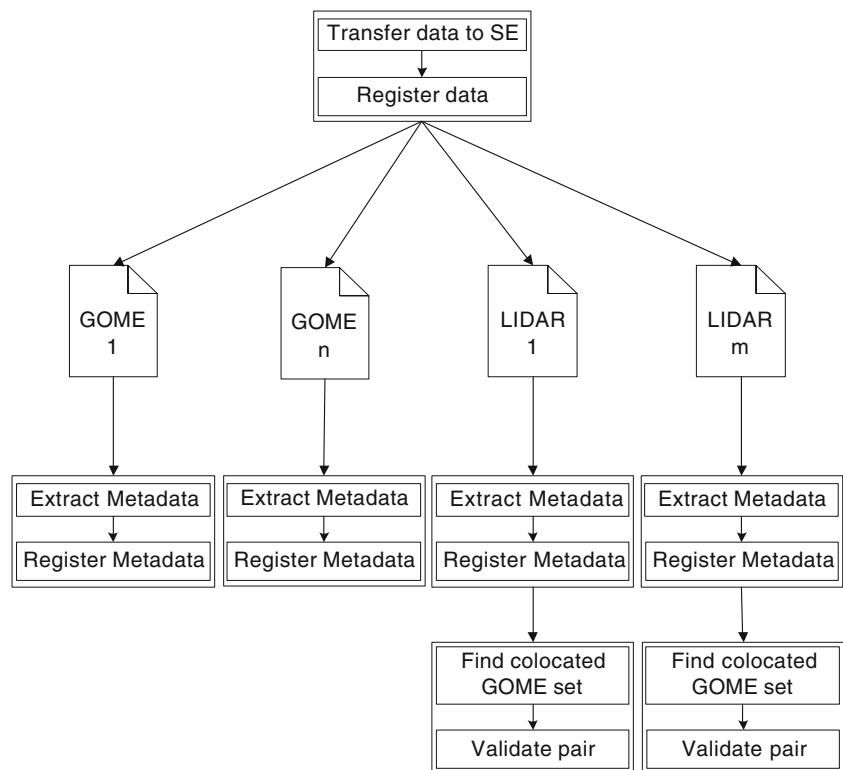
A complete list of requirements related to this test suite is listed in appendix I.

Test description

Objectives of the test

The main objective of the ES-TS-GOMEVAL test suite is to examine the middleware for the requirements mentioned above. But the reliability, robustness and speed of the data and database access is tested too. The performance of the three components is tested. Although interesting, semantic mismatching between ES and Grid data management systems and services are not part of the test. The way to

Fig. 3 GOME Validation workflow with n GOME files and m LIDAR files



test semantic mismatch would be to do a formal vocabulary comparison and/or user testing to see when they were not able to use the grid system because on lack of understanding of the terminology.

Part of the test focuses on the interface between a RDBMS server and the Grid infrastructure. This interface needs to support a database which allows geospatial searches with concave polygons (orbits). PostgreSQL with the PostGIS extension provides a possible RDBMS.

Test workflow

The workflow for the GOME data validation as presented in Fig. 3 is used for the test. It consists of the following steps.

1. Transfer the data (GOME and Lidar) to a storage element on the Grid infrastructure
2. Register the data
3. Read the content of the data files to create the metadata tuple
4. Enter the tuples in the metadata base
5. Geospatial search of the collocated GOME and LIDAR observations
6. Run one validation job for each LIDAR file, with all collocated pairs of GOME files

Authorization for processor usage and data access

The GOME data needed is a free-of-charge product for scientific research (CAT 1 product). Registration at the ESA Portal (<http://eopi.esa.int/esa/esa>) is required to get access to the data. The NDSC Lidar data have only constraints for publication and must be referenced when their data is used in publications. The GOME test suite needs 92 GB of disk space on storage elements for the GOME data. Table 2 provides the number and size of the different files for 1-year of data.

Pass/fail criteria

The test suite is successful if for all Lidar files all corresponding GOME data can be determined and validated. This means the test is successful if for each GOME profile none, one or more corresponding Lidar files can be listed and the result of the validation with these files is accessible.

All test cases have individual pass/fail criteria, which are specified at the individual test case. A test case is one step in the workflow described earlier.

Test Environment setup

The test environment setup consists of two steps:

1. Install and setup an RDBMS which support spatiotemporal queries, e.g. PostgreSQL with the postGIS extension
2. Install and setup the database interface

How to install PostgreSQL and the postGIS extension is described at their project homepages ([Http://www.postgresql.org](http://www.postgresql.org), <http://postgis.refraction.net>)

How to install and setup the database interface cannot be described here. The solution depends on the Grid middleware. One database needs to be created for the test cases and at least four users have to be created. One user who has admin rights on the database, one user who has only read and write rights on the database, one user who has only read rights on the database and one user who does not have read or write rights on the database.

Evaluation of grid data access tools with the GOMEVAL test suite

Introducing the tools

The software components we chose to test are OGSA-DAI (Antonioletti et al. 2005), AMGA (Santos and Koblitz 2006) and GReIC (Fiore et al. 2008).

OGSA-DAC (Open Grid Services Architecture—Data Access and Integration project) consists of a Globus Grid Service and development framework to query arbitrary data sources through a web service host and supply results and operations on results to Grid clients using the Grid Security Infrastructure (GSI, <http://www.globus.org/security/overview.html>). It is written in Java and can be deployed to a Globus or Apache Tomcat Server. It then offers a set of services for managing resources that handle communication to the real data sources. Queries in the native language of the data source (e.g. SQL for relational data bases) are embedded in requests to one of these services which can also execute operations on the responses. This way, an OGSA-DAI request can contain a kind of data centric

Table 2 Number and size of the GOME ozone profile and Lidar files

Dataset	Number of files handled (per year)	File size	1 Year
GOME	4,724	19.5 MB	~92 GB
Lidar	12	2.5 MB	30 MB

workflow of operations like queries to different data sources, conversion or merging of result sets, as well as delivery to different targets (e.g. Grid-FTP, FTP) which the OGSA-DAI service can carry out asynchronously. The API facilitates the construction of such requests by offering pre-defined objects for management and execution. It is also possible to design custom data source services, which can for example carry out static queries or re-occurring operations, but also to query more special data sources, like remote web services or web databases.

AMGA (ARDA Metadata Catalogue Project, <http://amga.web.cern.ch/amga/>) is now included in the EGEE middleware gLite as the default metadata database. It runs as a system daemon accepting connections through a SOAP interface or raw sockets using an own ASCII protocol. Queries to the data source are written in an AMGA specific query language which is internally parsed and translated to the language of the target system. For queries to SQL databases it connects through an ODBC interface and thus depends on the availability of ODBC drivers. AMGA is different from its competitors in that it offers a different view on the meta data. In AMGA, all metadata is attached to a file system-like hierarchical structure composed of collections (folders) and entries (files), so that every entry is identified by a complete path. Both of these entities can have properties of different types attached to them, which can be dynamically added or removed, which makes it a very flexible system. For transfers of larger volumes it reaches high throughput rates with its streaming implementation. AMGA also offers a sophisticated replication mechanism, which was not evaluated in this study.

GRelC stands for Grid Relational Catalogue and is a set of components offering integrated management and access of databases in the Grid. The Data Access component (GRelC DAS) runs as a system service offering connections through its C SDK for which proxy classes for C++ and Java are available. It internally connects to the database by the use of the specific client libraries that are offered by the database distributors, e.g. libmysqlclient by Sun. GRelC does not focus on the pure database query facility but additionally on the real integration of databases into the Grid environment, with integration to Information Services, Virtual Organizations, etc. It supports both classical relational data bases like PostgreSQL and MySQL as well as XML data bases like Xindice and eXist.

Methodology

To be able to compare the different approaches of the respective solution for the GOMEVAL test suites we designed the test implementation to use an interface to the metadata database, called MDBackend. For each of the solutions we implemented this interface such that we can

change the applications database. OGSA-DAI and GRelC can share the database where the metadata is stored in, because they both rely on raw SQL and do not influence the schema used. For AMGA we used a different database schema, because of its different perspective on metadata. To be able to exchange the backend without changing the application logic we used a data identifier that is available in both systems. We chose the logical file name of the original data file in the storage system. In the OGSA-DAI and GRelC databases this was stored as a String primary key of the metadata table, whereas in AMGA we devised a custom directory structure grouping the data by type. Each entry has its logical file name (LFN, a unique identifier of a file in Grid space) attached as an attribute while its entry name corresponds to the filename (base name).

The actual correlation of corresponding data sets is left to the underlying PostgreSQL server by using the PostGIS extension. Storing the coordinates of the measurements as PostGIS data types (Multipolygon / Point) allowed for queries selecting by spatial containment. For OGSA-DAI and GRelC the original SQL query as proposed in the test suite specification was used. For AMGA, the *CONTAINS()* function is available as an (at the time of writing undocumented) extension to the AMGA Query Language, if PostgreSQL is used as the backend. As the programming language for our tests we chose Java, as there are Java APIs available for all three projects. It is also the only language for which OGSA-DAI offers a client library. AMGA offers three different levels of abstraction of which we used the low level version, because of the PostGIS function we needed to apply to some of the Queries. To work with GRelC we used the GRelCJProxy library, which is available separately.

Results of the GOMEVAL tests on database access tools

Our tests showed that all three components are well suited for applications like the GOME validation application. All of them also show different specific weaknesses though. Table 3 summarizes the different functionalities available in the three data access tools.

While OGSA-DAI's strength lies in its ability to include arbitrary data sources by supplying DataSource services, it has a steep learning curve because of its service oriented architecture and advanced workflow capabilities. When we first implemented the test there was no support for Virtual Organizations. This should already be solved by OMII Europe (The Open Middleware Infrastructure Institute for Europe, <http://omii-europe.com/>). The application server and SOAP stack also impends a small overhead compared to the system services and raw protocols of the other two solutions.

AMGA on the other hand is very fast and easy to use with its rich API. It is very well implemented and integrates

Table 3 Functionalities of the OGSA-DAI, AMGA and GReIC data access interfaces

	OGSA-DAI	AMGA	GReIC
Objective target	Integrating arbitrary data sources	High-performance metadata catalogue	Grid database access and management
Relational DB	X	X	X
XML-DB	X	–	X
SQL/XQuery	X	Proprietary query language abstracting source languages, SQL is w.i.p.	X
gLite integration (BDII/VOMS)	w.i.p.	X	X
Grid security (GSI)	X	X	X
Access control	In backend (user mapped)	X	X
Custom data sources	X	–	–
DB replication	–	X	X
Streaming	X	X	X
Interfaces	SOAP	SOAP (WS-DAIS w.i.p.)/own protocol	SOAP/own protocol
Client API	Java	C++/Java/Python/Perl	C/C++/Java

into gLite on authorization and authentication level, with support for VOMS (Virtual Organization Membership Service, <http://edg-wp2.web.cern.ch/edg-wp2/security/voms/voms.html>). To restrict unauthorized access to data it supports access control lists. But its special way of presenting data and the uncommon query language it uses leads to an elongated preparation period and possibly impedes the reuse of previously collected data. It also means that applications already using databases need to be ported from using SQL to using the AMGA query language. We did not evaluate exhaustively whether the language offers the same expressive power as SQL, or if it restricts more advanced features. At least its way of tokenizing and translating the query string to SQL implies that custom language constructs, such as additional functions added by database extensions like PostGIS, need to be added to the parsing grammar and thus require a source code change and recompilation.

At the time of our tests, GReIC was not yet as sophisticated as the other contenders. It was weakly documented and needed a special version of PostgreSQL, gLite (<http://glite.web.cern.ch/glite>) and the associated Scientific Linux (<http://www.scientificlinux.org>) to install cleanly. It has since caught up and is moving fast and tests are still ongoing. Its API is straight forward and thus implementing the backend was without problems. It has other strengths too, like its tight integration with gLite, advertising and distributed databases, which shows its broader focus.

In a more abstract sense, our test showed us that contradicting requirements for such a development aggravate the decision for a universal solution. On the one hand, a solution compatible to classical application development is desirable and lowers the entry barrier for application porting. But the novel environment of the Grid demands

solutions for new challenges faced while porting or implementing applications. The structural properties of metadata are one such example. While it would be advantageous to use standard metadata schemas in dedicated data stores, it is also desirable to have them integrated into the Grid middleware for data file management, while at the same time being compatible to already existing custom data bases and code. The specific needs of an application regarding performance (be it latency or throughput), reuse of data, programming language requirements or data access policies all influence heavily the decision for the preferable solution.

General test suite evaluation results

The DEGREE project provided the test suites to several European Grid projects (11 in total). Besides the test suites also the Earth Science Grid Requirements document (Som de Cerff et al., 2007) was provided. Most of the grid developers did not try the test suites for different reasons, however some did participate. The summarised results are discussed in next paragraphs. Where applicable, the project teams and the test suites they evaluated are mentioned.

The Grid5000 project (<https://www.grid5000.fr>) tested the ES-TS-CMT test suite. Grid5000 develops software and tools. One of the tools, DIET, enables a large number of jobs to run simultaneously. The Laboratory de l'informatique du parallélisme (LIP-France), was contacted and the test done in December 2007. The requirements of the CMT test suite are based on the return of a collection of jobs sent simultaneously and job re-submission, when a job is not returned. DIET/Grid5000 seems to be a solution. LIP ran a first set of tests in December 2007. They submitted simultaneous jobs on different grid nodes, and checked their return. Up to that

point satisfactory results were obtained. The major difference with EGEE is that the CPU reservation is done before sending the job, that avoids job failure by site momentarily not available or too long queue. However the recovery after an execution failure was not tested. LIP was also interested in the workflow of this application.

EGEE (partly) tested ES-TS-GOMEVAL. The GOME test suite was performed by FhG/SCAI in the framework of EGEE (see also results in the previous chapter). The technical board of EGEE stated their interest in well documented use cases, which run daily or by SAM (Site Availability Monitoring). In response to the board interests simplified use cases have to be generated from the test cases.

The ES-TS-FFSC test suite was tested by UISAV (Institute of Informatics, Slovak Academy of Sciences) in the int-eu-grid project. They developed the application and used the Knowledge-based Workflow System for Grid Applications (KwfGrid, <http://www.kwfguid.eu/>) workflow system. This test permitted to point out missing functionalities.

All test suites were analysed by the P-GRADE/gUSE project (<http://www.p-grade.hu>), which permitted the detection of gaps between provided and required functionality in this environment. The analysis showed that seven out of nine test suites can be deployed using the P-GRADE/gUSE workflow tool. The test suites have not actually been run on Grid. Some test suites need an extension for data handling with the P-Grade portal. The ES community has to analyze in detail the new version of P-GRADE/gUSE in order to determine what applications can take advantage of the properties of this environment.

The European Space Agency (ESA) developed a ‘Grid on demand’ portal (GPOD) that integrates all the needed different services and makes the validation easy and fast (Retscher et al. 2007). The ES-TS-GOMEVAL GOME validation test suite and the ES-TS-GRIMI 2 test suite applications were executed successfully on this dedicated ESA infrastructure.

All test suite and requirements evaluation results have been reported to the European Union in the ‘Progress Report on ES Family of applications and their Grid Requirements (D1.2)’ and can be found on the DEGREE website (<http://www.eu-degree.eu/DEGREE/internal-section/wp1/DEGREE-D1.2.pdf/view>). The requirement evaluation in this document is based on the test suite results and on document surveys of existing Grid tools and infrastructures. In total seven Grid infrastructures and 25 Grid middleware tools were analysed. Conclusion from this analysis was 60% of all ES requirements are fulfilled in available Grid middleware tools, but only about 10% of the ES requirements are met in a single Grid infrastructure.

From discussions with grid developers during the test suite evaluations and at presentations of results at different

conferences, it appeared that the requirements listed in the provided documents are too detailed and not generic enough for grid middleware developers. The implementation of specific requirements may limit the usability of the middleware and hinder its evolution. Besides requirements change in the course of time. Their remarks corroborate the conclusions made at the DEGREE meeting at CRS4 and at the third e-collaboration workshop for the earth science community at ESA ESRIN. These conclusions could be summarized as follows:

- The requirements provided are too much related to EGEE middleware and as a consequence not generic enough.
- There is a confusion between middleware and tool requirements. Some requirements expected from the middleware are fulfilled by tools and the reverse. The confusion arises from the diversity in middleware. Sometimes upper-level Grid tools are integrated in the middleware.
- The test cases in the test suites often are too specific, because they are based on specific applications and do not test all the different aspects of a requirement.

Based on this, a start was made to restate the ES requirements. This effort will be continued by the ES community, as providing requirements and test suites is important to help grid developers make design decisions and set priorities in their developments.

Conclusions

The test suites have been an opportunity to contact a variety of Grid projects and initiatives, to open discussions with the grid developers on the ES requirements and to provide material for testing. During the DEGREE project only few tests were carried out. Grid5000 tested only a part of the ES-TS-CMT test suite. Int-eu-grid tested the complete ES-TS-FFSC test suite. All test suites were deployed in a P-GRADE/gUSE environment. The metadata database of ES-TS-GOMEVAL was used to test different data access interfaces. These few tests clarified the gaps and/or solutions for the functionalities required by ES. EGEE proposed to use parts of the test suites (use cases) for running daily quality tests of the EGEE infrastructure. Also revised requirements were offered to the European Grid Initiative-Design Study (EGI-DS).

More specific conclusions are summarized:

- The test suites provide an excellent way of understanding requirements for both middleware developers and ES scientists, and provide valuable input for monitoring requirements. Many contacts have been established. Sixty percent of all ES requirements are fulfilled in

available Grid middleware tools, but only about 10% of the ES requirements are met by a specific Grid infrastructure (Som de Cerff et al. 2008). But neither a single grid infrastructure nor a single grid middleware supports all requirements. Useful solutions do not exist for all ES requirements; some are merely “proof of concept” quality solutions and need further development.

- Some requirements can only be satisfied by application developers and middleware developers together (e.g. license management).
- Requirements need to be rewritten in cooperation with the middleware developers. The DEGREE requirements are partially too specific. This work will be continued in EGEE-III.
- Grid developers should standardize the access to grid services so new toolkits could integrate services to decrease the gap in requirements for ES. Recently, many grid middleware producers have implemented the OGSA standards for (web) services (e.g. GT4). This allows for the construction of a SOA and the relatively easy addition of existing services (administration services, GIS services), new services or coupled services (atomic operations). It would be beneficial if the data management services can be easily integrated in the grid middleware.
- An integrated metadata catalogue tightly coupled with the file catalogues and file transfer is desirable, allowing for flexible metadata creation as well as data discovery and lookup.
- Spatial and temporal data must be better supported. There is some support by applications, but possibilities will improve when the grid middleware adds support too.
- Some QoS related required functionality can be achieved by integrated toolkits, but there is a lack of functionality for requirements regarding integrity, validity and timing of operations. Users of replicated data must be able to have full confidence in the validity and integrity of the replicated data.
- Standards for ES are defined in the framework of European and worldwide programs, projects, directives and initiatives such as SEIS, INSPIRE, GMES, GEOSS and OGC. An interface standard for Grid technology towards OGC services is defined within OGF (Web Processing Service, WPS). The ultimate objective is to easily interface the ES environments with the Grid environments.

ES should join and help grid standardization efforts by providing requirements, use cases and test suites.

Acknowledgments This work makes use of results produced by the Dissemination and Exploitation of Grids in Earth Science (DEGREE) project, a project funded by the European Commission (under contract number IST-2005-034619) through the Sixth Framework Programme. DEGREE brings together 10 partners in 7 countries to reduce the gap

between ES users and Grid Technologies. Full information is available at <http://www.eu-degree.eu>.

Appendix 1 List of requirements related to the ES-TS-GOMEVAL test suite

Security requirements being tested

The following requirements concerning certificates and users are more or less tested implicitly because they mainly involve the procedure of “joining” the Grid.

- ES-S-1 Clear and straightforward procedure for obtaining a certificate and joining the Grid
- ES-S-2 Unified, common security protocol, standard for all services and resources
- ES-S-3 Standard security protocols seamlessly integrated into all components as fundamental design feature
- ES-S-4 From user perspective, security mechanisms should be non intrusive, transparent and should present similar functionality, independent of the type of Grid resources being accessed
- ES-S-5 User may belong to any number and combination of VO(s)/group(s)/role(s)
- ES-S-6 User may possess a single certificate valid for all associated VO(s)/group(s)/role(s)
- ES-S-7 Private key stored only on users computer; proxy generation directly from the users computer
- ES-S-8 User able to specify the required VO/group/role attributes to be associated to the proxy certificate during single sign on

Concerning *VO management* we have the following requirements:

- ES-VO-1 Definition of groups within a VO
This requirement can be tested with the access policy test cases, if more fine grained test procedures are added, which reflect the group permissions.
- ES-VO-2 Definition of roles within a VO
Like ES-VO-1 this requirement can be tested with the access policy test cases, if more fine-grained test procedures are added, which reflect the role permissions.
- ES-VO-3 Assign to a user one or more groups and/or one or more roles
Like ES-VO-1 and ES-VO-2 this requirement can be tested with an extension of the access policy test cases.
- ES-VO-4 Web based VO database administrator interface
This requirement concerns the procedure of “joining” the Grid and will not be tested directly by this test suite.

- ES-VO-5 Administrator interface login via the user's certificate loaded in browser
Like ES-VO-4 this requirement concerns the procedure of "joining" the Grid and will not be tested directly by this test suite.

Workflow requirements being tested

- ES-WL-1 Method for jobs running on the worker nodes to retrieve attributes and values that were specified in the JDL
As the Grid jobs need access to input parameters specified in the arguments section of the JDL this requirement is implicitly tested by all test procedures that use Grid jobs.
- ES-WL-2 Improved error messages
This requirement is implicitly tested by all test cases, as readable error messages are needed to track down failures.
- ES-WL-3 Provision and enforcement of job submission quotas to prevent overloading the RB
This requirement will be explicitly tested by all test procedures that use Grid jobs, by querying the job submission endpoint before job submission.
- ES-WL-4 Monitoring of RB state and job submission timings with results dynamically published in a Grid information service, to enable user/application to choose RB service based on current performance and loading
This requirement will be explicitly tested by all test procedures that use Grid jobs, by querying the load of all available job submission endpoints and submitting jobs to the endpoint with the smallest load.
- ES-WF-1 Intelligent assistant interface to facilitate scientific workflows and domain applications
This requirement is needed to execute the whole validation workflow in the simplest case this can be the ability to execute DAG jobs such as the gLite workload management system (WMS). If only DAG job is used here this requirement can be directly tested.

Data and metadata requirements being tested

- ES-DM-1 Fast performance of replica catalogue operations, resistance to degradation under heavy load, whether due to increased data or user numbers
This requirement is implicitly tested as test procedures working on the whole set of test data produce load and these tests will probably fail if the requirement is not fulfilled.

- ES-DM-2 *QoS response times published by each replica manager service in the Grid information system, retrievable by the end user/application*

This requirement is explicitly tested by some test procedures by first querying the QoS value and comparing this with measured time.

- ES-DM-3 *All functions, e.g. creating replicas, file transfers, inserting, updating and retrieving replica catalogue information and metadata, must include proper security checks on the proxy-delegated permissions*

This requirement will be explicitly tested with a positive and a negative test procedure. The security checks need to support very fine granular access policies. This involves:

- Support for ACL's
- Users may have an arbitrary number of roles
- Roles need to restrict the access to the data in the following ways:
 - A role need to be valid only for a certain time period
 - Roles should be able to restrict the access to only a part of the data or metadata, e.g. a role may allow a user to read a part of the data, a part of its metadata and to write one special field of the metadata scheme

- ES-DM-5 *Data replication (i.e. data transfer and catalogue metadata registration) must be performed as an atomic operation*

- ES-DM-6 *Correct management of disk space quota checking integrated in the data transfer operations*

This requirement is explicitly tested by some test procedures by querying the disk space quotas before a file transfer. If the quota is utilized the transfer is done but should fail.

- ES-DM-7 *Advance, pre-emptive warnings when available space falls below threshold levels*

This requirement is implicitly tested, because if an error occurs during test execution that is related to missing disk space and no pre-emptive warning was issued the requirement is not fulfilled.

- ES-DM-8 *Automatic data integrity checks embedded in replica manager operations*

This requirement is implicitly tested, because if an error occurs during test execution that is related to broken files and no warning was issued the requirement is not fulfilled

- ES-DM-9 *Capability to handle (storing, searching, updating, retrieving) file numbers of the order of millions of files*

This requirement is implicitly tested as test procedures working on the whole set of test data produce load and

these tests will probably fail if the requirement is not fulfilled.

- ES-DM-11 *Register both the logical filename and the associated set of metadata keys (list of key=value tuples) in a single operation*
- ES-DM-12 *Retrieve full set of metadata tuples associated to a LFN in a single operation*
- ES-DM-13 *Capability to store millions of files per VO*
This requirement is implicitly tested as test procedures working on the whole set of test data produce load and these tests will probably fail if the requirement is not fulfilled.
- ES-DM-14 *Mirroring of metadata and databases*
- ES-DM-15 *Capability to retrieve data from legacy databases, using Grid certificate authentication and authorization*

This requirement will be directly tested, as it affects several steps of the described workflow. It describes the need for support of an interface such as AMGA [R17] or OGSA-DAI [R18]. Alternatively it is possible that the middleware includes a component that supports access to an RDBMS outside the Grid. This means that the middleware comes with clients and client-libs (api) installed on each worker node and a installable server package. In this way the provider of a database just needs to install and configure the server package and than the database can be queried from the Grid using the middleware clients and client-libs. For this test suite the server needs to support a RDBMS which supports spatiotemporal queries, e.g. PostgreSQL with the PostGIS extension. The best would be if any RDBMS with an jdbc/odbc interface can be used.

- ES-DM-16 *Tools to create, update, query and remove (temporary) databases. These databases can contain multiple tables for both data and metadata. Databases can be relational, object based or XML.*
This is an optional requirement that can be tested if the databases support spatiotemporal concepts and no external RDBMS is used like it is described in ES-DM-15.
- ES-DM-17 *Data discovery services shall be able to handle spatiotemporal concepts*
Is an optional requirement that can be tested if the metadata capabilities of the data discovery services are used instead of the external RDBMS.
- ES-MM-1 *RC integrated support for user or application defined metadata associated to logical filenames, supporting wide range of available data types (e.g., date, time, polygon, integer, alphanumeric, float)*
Like ES-DM-17 this optional requirement can be tested if the metadata capabilities of the data discovery services are used instead of the external RDBMS. With this also the results of the validation of a GOME file can be saved as an attribute of the file.

- ES-MM-3 *Tools to publish data to these data discovery services shall be provided and shall use certificates for authentication and authorization*
This requirement is partly included in the description of ES-DM-15. As for all communication with the metadata base authorization tokens, like Grid proxies, are used this requirement is directly tested by multiple test procedures.

System integration requirements being tested

- ES-SI-1 *The Grid middleware must provide applications with command-line commands, library embedded APIs and service based access*
This requirement is related to the requirement ES-DM-15, as it concerns the possible ways to access external databases. Therefore it is tested by the same test cases.
- ES-SI-3 *Standard mechanisms and approach to fault tolerance, error handling and recovery supplied to all middleware components*
This requirement is related to the requirement ES-WL-2 and will be handled in the same way.
- ES-SI-4 *Common standard (guidelines) for command line formats and APIs*
This requirement will improve the overall usability of the Grid middleware and can therefore checked by all test cases.
- ES-SI-5 *Standardization of error codes, error messages and error handling procedures*
This requirement is related to the requirements ES-WL-2 and WS-SI-3 and will be handled in the same way.

Information system requirements being tested

- ES-IS-5 *Schema to include info on all available services, e.g. resource brokers, replica catalogues, replica optimizers, metadata catalogues, etc*
This requirement is tested by all test procedures by first querying for available services of the needed type and then using the returned service.
- ES-IS-6 *To allow clients to determine whether a service is usable or not, the schema should support inclusion of current serviceability status for all published services*
This requirement is tested by all test procedures by first querying the status of the service to use.

References

- Antonioletti M, Atkinson MP, Baxter R, Borley ANP, Chue Hong NP, Collins B, Hardman N, Hume A, Knox A, Jackson M, Krause A, Laws S, Magowan J, Paton NW, Pearson D, Sugden T, Watson P, Westhead M (2005) The Design and Implementation of Grid

- Database Services in OGSA-DAI. Concurrency Computation: Practice Experience 17(2-4):357–376
- Babik M, Habala O, Hluchy L, Laclavik M (2007) Semantic Services Grid in Flood-forecasting Simulations. In: Computing and Informatics 26(4):447–463, ISSN 1335-9150
- Clévéde E, Weissenbach D, Gotab B (2009) Distributed jobs on EGEE Grid infrastructure for an Earth Science application: moment tensor computation at the centroid of an Earthquake, Earth Science Informatics., this issue
- Del Frate F, Iapaolo M, Casadio S, Godin-Beekmann S, Petitdidier M (2005) Neural networks for the dimensionality reduction of GOME measurement vector in the estimation of ozone profiles. J Quant Spectrosc Radiat Transfer 92(3):275–291
- Dziewonski AM, Chou TA, Woodhouse JH (1981) Determination of earthquake source parameters from waveform data for studies of global and regional seismicity. J Geophys Res 86:2825–2852
- Fiore S, Cafaro M, Negro A, Vadacca S, Aloisio G, Barbera R, Giorgio E (2008) “GRelC DAS: a Grid-DB Access Service for gLite Based Production Grids”. CBMS 2008:355–360
- Fusco L, Cossu R, Retscher C (2007) Open Grid services for Envisat and Earth observation applications. In: Plaza A, Chang C-I (eds) High Performance Computing in Remote Sensing. Taylor and Francis, Boca Raton
- Hambling B, Morgan P, Samaroo A, Thompson G, Williams P (2006) In: Hambling B (ed) Software testing: an ISEB Foundation, British Computer Society, UK, ISBN 1-902505-79-4
- Hluchý L, Habala O, Tran V, Gatial E, Maliska M, Simo B, Slizik P (2005) Collaborative environment for grid-based flood prediction. Comput Inform 24(1):87–108 ISSN 1335–9150
- Iapaolo M, Godin-Beekmann S, Del Frate F, Casadio S, Petitdidier M, McDermid IS, Leblanc T, Swart D, Meijer YJ, Hansen G, Stebel K (2007) Gome Ozone profiles retrieved by neural network techniques: a global validation with lidar measurements. J Quant Spectrosc Radiat Transfer 107:105–119
- Liakka J (2006) Validation of the dynamical core of the portable university model of the atmosphere (PUMA). Master's thesis, Uppsala University (ISSN-1650-6553 Nr 130)
- Retscher C, Fusco L, Zehner C, Fehr T, Casadio S, Loescher A (2007) Applications on a grid infrastructure for atmospheric instruments GOMOS, MIPAS, SCIAMACHY, and CHAMP. Proceedings of the ESA Envisat Symposium, 23–27 April 2007
- Santos N, Koblitz B (2006) Distributed Metadata with the AMGA Metadata Catalogue, in Proc. of the Workshop on Next-Generation Distributed Data Management (HPDC'15) Paris, France, June 19-23 2006
- Schwichtenberg H, Gemünd A, Kraus J, Rubin D (2007) Contribution to test suite and specification report (data management), DEGREE EU deliverable 1.3, available: http://www.eu-degree.eu/DEGREE/internal-section/wp2/DEGREE-WP2-D2_2-Test-Suite-1.3.pdf
- Som de Cerff WJ, Petitdidier M, Boonne C, Huchly L, Lonjaret M, Schwichtenberg H, van de Vegte J, van der Wel F, Vilotte JP (2007a) report on ES family of applications and their requirements, DEGREE EU deliverable 1.1, available: <http://www.eu-degree.eu/DEGREE/internal-section/wp1/DEGREE-WP1-D1-1.3.pdf>
- Som de Cerff WJ, Petitdidier M, Kraus J, Plevier C, Schwichtenberg H, Tran V (2007b) Contribution to test suite and specification report, DEGREE EU deliverable 1.3, available: http://www.eu-degree.eu/DEGREE/internal-section/wp1/DEGREE-WP1-D1_3-Test-Suite-Organization-v1.5.pdf
- Som de Cerff WJ, Petitdidier M, Gemünd A, Gotab B, Horstink L (2008) Progress report on ES family of applications and their Grid requirements, DEGREE EU deliverable 1.2, available: <http://www.eu-degree.eu/DEGREE/internal-section/wp1/DEGREE-D1.2u.pdf>
- Tran V, Habala O, Lonjaret M, Poliakov A, Schwichtenberg H (2007) Contribution to test suite and specification report (job management and control), DEGREE EU deliverable D3.2, available: http://www.eu-degree.eu/DEGREE/internal-section/wp3/DEGREE-WP3-D3_2-Test-Suite-d1.1.pdf
- van der ARJ, van Oss RF, Pijters AJM, Fortuin JPF, Meijer YJ, Kelder HM (2002) Ozone profile retrieval from recalibrated Global Ozone Monitoring Experiment data. J Geophys Res 107, doi:10.1029/2001JD000696
- Zhizhin M, Kihn E, Redmon B, Poyda A, Mishin D, Medvedev D, Lyutsarev V (2007) Integrating and mining distributed environmental archives on grids. Concurrency and computation: Practice and Experience 19(16):2157–2170, doi:10.1002/cpe.1192
- Zhizhin M, Kihn E, Redmon B, Medvedev D, Mishin D (2008) Space Physics Interactive Data Resources. Earth Sci. Inform. doi:10.1007/a12145-008-0012-5