

Data Requirements for The Grid Scoping Study Report (Status Draft)

CONTENTS

Contents	2
Purpose of Document	5
Related Documents	6
Summary	7
Introduction	9
Acknowledgements	9
Requirements Gathering	10
Data Structure and Representation	11
Data Source	11
Data Resource	11
Database	12
Data Formats and Precision	13
Data Classification	14
Data, Information, and Knowledge	14
Data Types	14
Raw Data	14
Reference Data	15
Processed Data	15
Result Data	15
Derived Data	16
Metadata	17
Technical Metadata	17
Location	17
Data Structure	18
Data Resource Characteristics	18
Contextual Metadata	19
Classification and Ontology Based Metadata	19
Derivation Based Metadata	19
Contextual Metadata Components	20
Ownership	22
Versioning	22
Provenance	24
Data Access Control	26

Data Publishing and Discovery	28
Information Requirements	28
Data Publishing Functionality	29
Data Discovery Functionality	30
Data Retrieval	32
Specifying the Target	32
Specifying the Output	32
Specifying the Retrieval Conditions	32
Data Analysis and Interpretation	34
Methods of Working with Data	36
Data Management	37
Data Volumes	37
Data Lifecycle	38
Data Management Operations	39
Appendix I Interview List & Sources	41
Appendix II Data Requirements Questionnaire	42
Appendix III Collated Requirements and Dependencies	52
Appendix IV Requirements Prioritisation	59

DOCUMENT CONTROL

Author: Dave Pearson, Oracle Corporation UK

Document Reference: DBTF/REQ/001

Document Version: Draft 1b

Effective Date: 08 Feb 2002

Change Record

Date	Author	Version	Change Reference
07-Feb- 2002	Dave Pearson	Draft 1b	Revisions and Appendices added

PURPOSE OF DOCUMENT

This document is a report on the findings of an analysis exercise conducted in the UK to identify the requirements for creating, maintaining, and accessing data in a Grid environment. It defines the scope of the requirements in generic terms, and describes each major requirement identified in a high level of detail. The report is not necessarily a definitive record of all potential requirements for data in a Grid environment, nor does it describe any requirement in sufficient detail to be a sole input for designing solutions.

The report is intended for use as input to define conceptual models that meet the requirements, and to plan a prioritised programme of work to refine the requirements in more detail, to prototype designs, and to implement useable software components. It may also serve as an introductory overview to the importance and role of data in Grid.

RELATED DOCUMENTS

The following documents are related to, or have provided input for the content of this report:

Research Agenda for the Sematic Grid, A future for e-Science Infrastructure, D De Roure, N Jennings, and N Shadbolt, December 2001

Databases and the Grid, P Watson, January 2002

Database Access and Integration Services on the Grid, N Paton, M Atkinson, V Dialani, D Pearson, T Storey, and P Watson, January 2002

SUMMARY

This report is not intended to be a definitive list of all Grid data requirements, and they are not described in sufficient detail to be the sole basis for designing solutions.

The key findings of the data requirements scoping exercise can be summarised as follows:

There is no single standard in any science discipline that defines the agreed representation, format, and structure of data. The Grid must be capable of supporting all data defined to any existing standard, and to individual standards by owners who wish to make their data accessible in a Grid environment.

Users may be prepared to make their data available in a Grid environment, but in some cases they wish to maintain control over who can access or change the data content. The requirements for access control demand that restrictions can be applied at almost any level of granularity over data content and user grouping.

The ability to define and maintain Metadata, 'data about data', is extremely important in the Grid. Metadata provides the key to making data more accessible through a service-based architecture, and to defining data provenance. It is also key to creating 'virtual databases', and to achieving levels of abstraction that enable users to access data content without needing to know its location or its internal structure.

Provenance is an essential requirement for all users who wish to discover and use data in a Grid environment. It is necessary for establishing the reliability and quality of data, and for recreating data and experiments accurately.

It would be wrong to assume that data in a Grid environment are always read-only. Changes to data occur during processing, analysis, and interpretation. Users are increasingly defining volatile data in complex data structures; in Database Management Systems and in emerging XML formats. This introduces the need for the Grid to ensure the integrity and consistency of data is maintained when changes occur.

Scientists, particularly in life and environmental science disciplines, increasingly need to integrate data from a wide range of sources, and to integrate data held in different structures and representations. The Grid must be able to maintain the context that this integration creates throughout workflows, and potentially throughout the entire duration of a study.

The current explosion in data volumes, brought about by the availability of affordable high performance computing platforms and high capacity disk storage, is expected to continue in all areas of science for the foreseeable future. The highest annual growth rates are currently in those science disciplines that are exploiting *in silico* experimental and simulation techniques.

The prospect exists for literally billions of data resources and petabytes of data being accessible in a Grid environment. The challenges for managing these volumes and varieties of data in a heterogeneous distributed environment are significant.

INTRODUCTION

This report contains the findings of an analysis exercise conducted to define the scope of data requirements for the Grid. The decision to conduct the analysis exercise was taken jointly by the UK Grid database and architecture taskforces. Both groups considered the exercise was needed because a documented scope and description of the principal requirements provides the basis for achieving a common understanding of needs, and of the dependencies between them. They also considered the findings would provide useful input for defining a conceptual model and a roadmap for meeting the requirements.

The report represents the first stage in a process that will ultimately refine the requirements and describe them in sufficient detail for Grid data services to be designed, and for reference implementations to be developed.

The report is structured into a number of sections. In each section, the findings are presented as a set of generic requirements, and terms are defined that are commonly used elsewhere in the report. A full list of generic requirements, ordered by type is presented in Appendix III. It includes additional generic requirements and dependencies between requirements that have been inferred during analysis of the findings. These are identified in the listing. Some guidelines on prioritising the requirements are presented in Appendix IV.

Acknowledgements

The author is grateful to everyone who participated in the analysis exercise. Particular thanks are extended to those who participated in the interviews, completed the questionnaire, and provided feedback and additional input during the review stage.

REQUIREMENTS GATHERING

The requirements described in this report were gathered in the course of an analysis exercise carried out over three months, during the period November 2001 and January 2002.

The analysis exercise used interviewing and questionnaire techniques to gather requirements. Interviews were held and questionnaire responses were received from UK Grid and related eScience projects. A full list of the projects involved in the exercise is given in Appendix I, and the questionnaire is reproduced in Appendix II. Additional input to the requirements has come from a number of published sources, and from discussions and presentations. The information identifies elements of the requirements at CERN in Geneva, and the European Astrowise and Datagrid projects. The sources are listed in Appendix I.

The interview notes, questionnaire responses, and additional information were collated into a set of generic requirements, and an initial report was delivered to the UK Grid Architecture Task Force, and to interviewees and respondents during December 2001. The schedule of interviews was completed in January and feedback from the initial report was consolidated. A summary of the collated requirements is presented in Appendix III.

DATA STRUCTURE AND REPRESENTATION

This section defines terms that are used throughout the remaining sections of this report. It also describes the structures and representations in which data occur and that the Grid must support.

Data Source

A data source is an instrument, a device, or an application program that creates and outputs data. A data source need not be connected to the Grid infrastructure, or defined in a Grid environment in order to output data.

Astronomical telescopes, digital and video cameras, X-ray diffractometers, detectors in particle colliders, and remote sensing devices are all examples of data sources. Programs that perform *in silico* experiments or run simulations of models are examples of application data sources. The duration and frequency over which a data source creates output is normally determined by experimental protocol, or by the requirements of a study or survey. For example, the LHC experiment at CERN will generate output over a number of years.

Instruments and devices normally output raw data, but subject to hardware being configured they may output processed data, and may store the output locally. A data source may create real time or pseudo-real time output, and it may stream the output directly to a display device for viewing. However, it is not implicit that any or all of the output created by a data source will be captured and stored in a persistent state.

The Grid must provide the ability to capture output directly from a data source that is connected to a Grid environment, and it must provide the ability to import output from a data source that is not connected. It must also have the ability to integrate output with existing data in a Grid environment, e.g. historical processed output, and make the combined result data available for pseudo-real time analysis and interpretation. This type of ability is necessary to enable dynamic monitoring and control of other types of Grid resources.

Data Resource

A data resource is a persistent data store in a Grid environment. It has an owner and a name, and is stored at a physical location in a file system or in a raw device. There are no constraints over type, structure, volume, or status of the content a data resource can hold. A data resource may conform to an agreed standard, or be totally owner defined.

It follows that to have any value and meaning in a grid environment, a data resource should be readable by someone other than the owner. Users can be granted

privileges to read all or part of its content, to create new data, and to modify and delete data content.

The Grid must support any type of data resource, and provide the ability for data owners and custodians to manage data resources online. They must be able to copy and replicate data resources across multiple sites in order to satisfy the service level requirements defined for a Grid environment. They must also be able to manage the archiving and recovering of data resources to and from offline.

Database

For the purposes of this document, a database is defined as an organised collection of data. The term database does not implicitly mean data held in a database management system (DBMS).

The collection of data may span one or more data resources, and may contain one or more data types. Invariably, the contents of a database are linked in some way, usually because the data content is common to a subject matter or to a research programme. Examples of databases include a centralised repository of gene sequences, and an individually owned directory structure containing spreadsheet data files of gene sequences.

The ability to group a logical set of data resources stored at one site, or across multiple sites, and to be able to name them as a single entity is an important requirement, particularly for curated data repositories. It must be possible in a Grid environment to reference a 'virtual database' and to perform set operations on it, e.g. distributed data management and access operations. This must apply when all or part of a virtual database is maintained centrally at a nominated storage site, or held locally on a PC that is known within a Grid environment. An example of a virtual database is the virtual observatory currently being developed by the Astrophysics community.

The structural organisation of a database reflects the logical groupings of each type of data held in its collection. The organisation of data held in a DBMS tends to be highly structured and the logical groupings must conform to a defined data schema. The grouping can be based on fields, records and files. In a relational database management system (RDBMS), the grouping is based on columns, tuples(rows), and tables, and in an object oriented database (OODBMS) it is based on objects and classes.

Data owners frequently choose to use database management systems because they provide a number of important facilities for managing data. These facilities can automate; controlling access to data, managing the referential integrity of data within transactions, logging changes made to data, auditing database activity, synchronising data replicated in a distributed environment, and recovering data to a consistent state. Database management systems also provide facilities to reorganise database contents in order to manage space efficiently, to optimise queries, and to balance available resources dynamically.

Document Control

Semi-structured data is defined using XML, a mark-up language that allows data to be described in user extensible ways. XML offers great flexibility in how the structure, and the syntactical and semantic rules of data are conveyed, and how structural links and non-linear pathways between data can be defined.

Database management systems and XML are capable of representing complex data structures that reflect naturally occurring and user-defined associations in the data. The ability to create user-defined associations is particularly important during interpretation when inferences about correlations can be recorded and linked back to the source data. It is also important for recording provenance.

Databases that hold semi-structured data defined in XML are becoming increasingly important in Science, particularly for curated databases that are published and distributed across multiple sites, e.g. gene sequences. XML is also being increasingly being used to define declarative simulation models, e.g. in neuroscience and in ecology.

Data Formats and Precision

The Grid needs to accommodate every type of data format and representation, and it must be able to support mixed types when they occur in the output of data sources and in the content of data resources.

Numeric data occurs in a wide range of formats and precisions, and the Grid needs ability to maintain some numeric data at the highest levels of accuracy. This is specifically the case with HEP, Astrophysics, and Engineering data. Textual data occur in many different formats and structures, and the Grid needs the ability to hold and perform operations on textual data in different languages, ontologies, and coding systems. For example, digital libraries may contain papers published in more than one language, and text information may not always be held in Ascii format.

Audio and multimedia can also occur in different formats and structure, and the Grid needs the capability to maintain this type of data in user-defined formats. For example, the Astrophysics community has standardised on its own image format termed, FITS.

DATA CLASSIFICATION

This section classifies the principal types of data that can occur in a Grid environment. It describes each type of data type in terms of the requirements that must be met. It also defines terms that are used throughout the remaining sections of this report.

Data, Information, and Knowledge

Data is a collective term for the values assigned to data items created by a data source, and to values assigned to items stored in a data resource. Information is a data item value that has context. For example, the values 5, 10, and 20 are data. Precipitation measurements of 0mm, 2mm, 250mm, and 2mm that represent the average quarterly rainfall in an area over a period of 5 years can be considered to be information. Through applying experience, knowledge about the climate can be derived resulting in the area being classified as a *desert*.

One of the problems in distinguishing between data, information, and knowledge in science is that one researcher's knowledge can become information or data when used by another researcher. For example, a biologist might simply use the knowledge in the previous case, *desert*, as a data input parameter for a simulation run of an application modelling biodiversity. Consequently, for the purposes of scoping the requirements, this document uses data as a generic term to refer to all three definitions.

Data Types

Data in a Grid environment can be classified in a number of ways. For the purposes of this requirements document, data classification is based on a combination of creation, purpose, and usage. The types of data in this classification are; metadata, raw data, reference data, processed data, results data, and derived data. Within the Grid there are no constraints over what formats and representations can occur in a data type. Metadata is described separately in a following section.

Raw Data

Raw data are created and output from a data source, either an instrument or an application program. The structure and format of raw data are determined by the data source. Instrument data sources tend to create raw data with limited value until processing has taken place. Examples of raw data include event data from collider detectors, meteorological data from remote sensors, astrophysical data from optical surveys.

Reference Data

Reference data are values that are frequently referenced in other types of data, and in processing, transforming, analysing, annotating, and interpreting data. Common types of reference data include; standardised and user defined coding systems, parameters and constants, and units of measurements. Examples of reference data include; the calibration constants for a scientific instrument, and the constants for a map projection or a coordinate system.

A feature of all types of reference data is that their data values remain static or change rarely. Although classification systems and ontologies are defined as metadata, their shorthand codes are a type of reference data.

Processed Data

Almost all raw data undergo processing of some form. Processing applies the necessary corrections and calibrations, and transformations data into meaning units of measure. It may filter out data that fail to meet the required level of quality or integrity, and data that do not fall into a required specification tolerance. For example, the great majority of data is lost in the initial processing of event data created by detectors in a collider because the data contain no anomalies. Conversely, processing may include merging and aggregation of data from other sources. It may also add additional processing specific information for provenance purposes. Processed data can be the result of several stages of processing, and raw data can be subjected to repeated processing. This can occur on a periodic basis, e.g. whenever calibration parameters are refined, and when processed data content is migrated to a new format. Reprocessing can also result when issues arise over the provenance or quality of processed data.

Result Data

Result data are created as the output of a data retrieval or interrogation operation, normally within an application or when examining data content during the discovery process. The types of application functionality that produce result data output include algorithmic analysis, simulation, and data transformation. Output created by data interrogation is always based on existing data. It is a subset of content from one or more data resources that have satisfied the specified selection criteria. The interrogation process includes data merging where more than one resource is involved.

Typically, a result data set is extracted from a database for the purpose of subjecting it to focused analysis and interpretation. It may be a statistical sample of a very large data resource that cannot be feasibly analysed in its entirety, or it may be a subset of the data with specific characteristics, e.g. event data from HEP experiments, and gene expression data. A result data set may also be used as input data for a simulation run in a modelling application. It may also be a set reference data for a visualisation application, e.g. map projection reference data, and oceanographic data for a regional

Document Control

study. User may choose to create a copy of the result data and retain it locally for reasons of performance or availability.

Derived Data

There are two main ways to create derived data. The first is by performing statistical analysis on other data to create statistical parameters, summarisations and aggregations. This type of data can be considered to be a form of results data and it is particularly important when analysing trends and correlations in data. It frequently comprises a significant element of the data in a data warehouse.

A second way to produce derived data is through recording observations on or drawing inferences from other data, or any subject under investigation. The definition of subject is very broad and includes, image, scientific sample, and environment. An observation is an annotation or a description of the features, properties or behaviour of data, experiments, or subjects. The need to record annotations is a common requirement in science. For example, it is required for describing the features in gene sequences, the properties of molecules, and environmental conditions in an experiment.

An inference is a deduction or conclusion drawn from analysis and interpretation. Inferences add to understanding and knowledge, and they are used to explain correlations, trends and anomalies. They may also include evidential reasoning. For example, in an environmental study, an inference can record that the death of fish in a river is correlated to levels of concentrations of toxic waste exceeding specified threshold. Evidential reasoning may record that the fish die when the level is lower than a threshold identified for a different species of fish in a separate study.

Inference data can be volatile because understanding may change, and hypotheses may be refined over the course of a study. Equally, inference data may not always be definitive, particularly when inferences are drawn about the same corrections and anomalies but are recorded by collaborators with different opinions. For this reason it is important that the Grid provides the ability to maintain personalised versions, and multiple versions of inference data.

METADATA

This section defines the requirements for metadata, the role metadata performs in data operations, and in establishing the quality and reliability of data.

Metadata is the term for 'data about data'. It is structured information that characterises the data it describes by conveying context and meaning. For this reason metadata adds value to data. The ability to define and reference data through metadata was considered to be an essential requirement by everyone involved in the analysis exercise.

Metadata is essential to meeting many other data requirements in the Grid. It is essential for facilitating data management tasks that are involved in maintaining the integrity and consistency of data, and for tasks involved in publishing and discovering data. Metadata is referenced when performing processing, retrieval, analysis and interpretation of data, and it is important for establishing the ownership, currency, validity, and quality of data. Metadata is also essential to the development of Grid services because it enables data operations to be abstracted to a sufficient degree that services can be created and made reusable. This facility makes it possible to access and manipulate data content without knowing where it is physically located, or how it is structured.

The Grid needs to provide the capability to define and maintain several types of Metadata in order to meet the data requirements identified during the analysis exercise. These types are technical, contextual, currency, and ownership metadata.

Technical Metadata

Technical descriptions and characterisations of data resources are required for all data source and resource related operations in a Grid environment. The information that describes and characterises data in technical terms are:

Location

The location defines where a data source or resource is located. It must be possible to define a location as a physical address, and as a logical reference to a data resource. An example of a physical location is the sector address on a raw disk partition. Examples of logical locations include; a full file pathname, a url, and an object name in a database management instance. Data sources also have a location, and if an instrumental data source has local storage it also has a data resource location.

Data Structure

The data structure defines the logical groupings of data items and their associations in a data resource, together with their order of appearance, format, size, and type within each logical grouping. It also defines the output of a data source in the same way. A definition of data structure is required to enable a user to access data directly, and to navigate through the content of a data resource. The data structure of a file is commonly termed the record structure, and the data structure of a database management system is commonly termed the data schema.

The Grid must provide the capability to define fully the structure and associations of data items for all types of data resource; including non-standard structures. This is particularly important for maximising access to existing data, often held in user defined file structures. It is also essential that the facilities to define metadata are extensible to accommodate new data types as they evolve. Examples of data structures identified during the analysis exercise include; user-defined files with mixed formats and representations, XML semi-structured data files, spreadsheet work files, and OODBMS and RDBMS data schemas.

Data Resource Characteristics

Data resource characteristics define information that is important in determining the most effective and most efficient methods for managing, discovering, and accessing data resources.

The size of a data resource is important information when deciding; where and for what duration data may be stored, where a resource might be replicated, or what local storage must be pre-allocated before copying a file. It is also important for determining what time and cost may be involved in accessing data at remote sites, and in duplicating data resources across multiple sites.

Access paths to the contents of a data resource, together with size and the number of logical records/objects/tuples in a data resource are necessary for enabling users to determine how best to access data. For example, in some cases it can be more efficient to scan a data resource than to access the majority of its content through indexed pathways.

It is important to know data volumes and access paths when accessing large data resources, and for load balancing other types of resources in a Grid environment. Consequently, the Grid should provide facilities to capture this type of information automatically when data resources are created or modified.

The status of data content in a resource is important information for establishing its quality and reliability. It can also be useful for determining access rights to data. For example, when the data content has a provisional status it may prevent the resource being placed in the public domain.

Contextual Metadata

Contextual metadata conveys meaning and context. In the Grid, there is a requirement for two types of contextual metadata; that based on a naming classification or ontology, and that based on derived data.

Classification and Ontology Based Metadata

The data descriptions in classification and ontology based contextual metadata conform to a set of agreed naming conventions, or terminology, and the data structure and associations conform to the syntax and semantics of the classification or ontology. Consequently, this type of contextual metadata is nearly always subject domain specific, e.g. gene ontology and a chronostratigraphic classification.

The use of structured naming classifications and ontologies is well established throughout science, and also in many industrial and commercial environments. However, the degree to which classifications are standardised and are accepted is very variable. There are very few instances in any scientific domain where one classification is universally accepted and employed as the only standard.

The value of classification based contextual metadata is that it provides a method for defining the meaning of data accurately and unambiguously. For example, the name *Homo sapiens* distinguishes it from all other species, and its position in the classification conveys a significant amount of information about its characteristics, and about its relationships to others species. The Grid must be capable of supporting all types of classifications and ontologies, including those with the most complex structures and associations; e.g. chemical, bioinformatics, and environmental data, and medial therapies.

Derivation Based Metadata

Derivation based metadata is used to specify the name of a data value that is derived from other data, and to provide a brief description of its derivation. For example, *average- daytime-temperature* may have a description that explains its method of calculation. This type of contextual metadata is commonly used in data warehousing applications when it is more efficient to store derived data than to recalculate the values dynamically each time they are required. There are many instances in science and engineering when it is advantageous to define derived data context. For example, the metadata name *mean-time-between-failure* may apply to a value derived from millions of data readings taken on an aircraft engine's performance over a long period of time. The data readings may be held across a number of sites, at each location where the engine has been serviced. It would be unrealistic to recalculate this value each time it is referenced in an interactive environment where response times are critical. It would also be time consuming and resource intensive to do so.

The ability to describe data content through contextual metadata provides a number of important benefits for Grid users. First, the contextual metadata conveys meaning about the data content. This is important when discovering data and understanding its content. Second, quality of data described through contextual metadata is enhanced because the descriptions conform to a set of agreed rules; either structural relationships or processing derivation. This reduces the possibility of ambiguity arising when interpreting and analysing the data. Third, contextual metadata allows users to reference data in logical terms that they understand. This means users do not need to know the underlying data structures or schemas of the data they are accessing.

Contextual Metadata Components

The components of contextual metadata that the Grid needs to provide to meet the requirements fully can be summarised as follows:

Classification, Ontology

The classification or ontology name and a description are required for publishing and discovering data. The name is required for specifying which terminology will be used when data content is referenced through contextual metadata. A description of the classification is required to assist users in assessing the suitability of data content when it is discovered. When several classifications or ontologies exist for the same subject matter, the ability to mark one as the preferred definition or preferred terminology is required.

Data name, Code, and Description

The data name, or term, is the accepted terminology for data item in a classification or ontology, or for the name of derived data. In some classifications, the name may be a compound name based on two levels in the hierarchy; for example, genera and species. The rules for a classification may permit alternative names for the same data item, i.e. synonyms, and therefore it is a requirement to be able to mark a data name as the preferred term. For example, one brain atlas ontology has three names for the same area of the brain. A description of the data name is required to assist users during data discovery, and in understanding the source and processing of derived data.

The data name provides a means to refer to an item in a data structure or schema without knowing its physical name. However, because naming conventions commonly result in lengthy names it can be convenient to refer to data using shorthand codes. The Grid needs to provide the capability for users to reference data by either means.

Classification and Ontology Structure

The classification structure defines the associations and dependencies between data names in a classification or ontology. The ability to record the structure is required to enable a user to interrogate the associations in a classification or ontology, and to navigate through its hierarchy. This is important when discovering and browsing data, and in establishing its suitability for referencing data content.

Mapping

Mapping defines equivalence between terms in related ontologies or classifications. The ability to map relationships between ontologies is particularly important because of the lack of agreed standards in terminology across all science disciplines.

Contextual metadata mapping enables users to compare classifications and ontologies in terms of their naming conventions, and structural relationships and rules. It also enables them to establish what alternative definitions are available for referencing data content.

The ability to map contextual metadata to physical data structures and schemas is a requirement in order to enable users to access data content using logical references; i.e. without needing to know the definition of its underlying record structure or data schema.

Mapping, in conjunction with contextual metadata, enables users to integrate data sets defined in different classifications and ontologies. This provides the ability to specify a single set of search criteria and data matching rules when performing integrated or federated queries against multiple data resources, and for referencing data in a virtual database.

Rules

Rules allow additional conditions and constraints to be defined for data, metadata, and mappings. They can be used to define a range or list of valid data values that a term can take, or the permitted length of a code. They can also define the sequence of actions to be used to resolve conflicts in mappings, and to identify when deadlocks have been reached. Several rules may be defined for each metadata mapping, and at each level of granularity in mapping. For example, the naming classification for therapies used in patient records is normally country specific. However, the structure of the classifications is not always the same. In an international clinical trials study multiple rules may be required to navigate through classifications when comparing the medical histories of patients defined in different terminologies.

Ownership

Knowing who owns data is important for a number of reasons. It is necessary for the owner to establish intellectual property rights over data, and for users to credit an owner when using their data. It is important information during data discovery because ownership is an indication to the quality and reliability of data. Users may need to know who owns data in order to resolve any issues relating to data source and provenance, and if they need to seek additional access rights to the data content. Ownership is also important information for accounting purposes, particularly if a charge is levied for storage, or for accessing data within a Grid environment.

Versioning

The ability to version data was identified as an essential requirement in the analysis exercise by all of the participants. Versioning provides the ability to distinguish between different states in the content of data over time, and the ability for different states in data content to coexist in the same environment, e.g. multiple annotations of the same gene sequence.

Versioning is important when data changes in any way, either by amending or deleting existing content, or by adding new content. For example, engineering designs and specifications can change over time, but is always important to know the exact configuration of an engine when it is being maintained. It is also important when data can be subjected to repeated creation, processing, annotation, and interpretation. For example, when reprocessing arises from periodic refinement in the calibration of an instrument data source.

Versioning provides a means of defining the currency of data, and of preserving a history of changes made. Consequently, it is an essential component of provenance. It enables multiple states to be retained online and identified, and it enables historical states to be recovered or recreated from archive. Versioning also facilitates data management, particularly when data is replicated or copied across multiple sites. It enables the consistency of data to be maintained, and it enables new versions of data to be replicated to sites before they are made available for use.

All types of data in the Grid can be subjected to versioning. Versioning can be used to distinguish subsets of raw and processed data that have common characteristics, or have been filtered according to different criteria. It can also be used to distinguish the result of reprocessing raw data using different calibration parameters. This is particularly important when calibration parameters are refined through improved understanding of instrument behaviour.

Versioning is important in the analysis and interpretation stages of research. It can be used to distinguish multiple interpretations of the same data, and different stages in the maturity of interpretation. It can be used to distinguish between different releases of documents, programs, and software, and between different states of models and simulations.

Document Control

Versioning is an essential requirement for managing multiple mappings of metadata at the logical to logical, the logical to physical, and the physical to physical level. An important implication of metadata mapping is the requirement for versioning to be applied at different levels of granularity, and that version changes may need to be propagated through a hierarchy when any definition in a mapping changes, or when the content of any definition in the hierarchy changes, e.g. when a component in the product breakdown structure of engineering design is modified.

PROVENANCE

This section defines the term provenance. It also describes the key requirements for creating a record of data provenance, and the role provenance plays in all data operations.

Provenance, sometimes known as lineage, is a record of the origin and history of a piece of data. It is a special form of audit trail that traces each step in sourcing, moving, and processing data. It can apply to a single data item, a logical data record, a subset of a database, or to an entire or database.

In science, one researcher's output can become another's input. This makes provenance an essential requirement in a Grid environment when a user chooses to use data created and maintained by someone else. Provenance is key to establishing the quality and reliability of data in publishing and discovery processes. It provides information that is necessary for recreating data, and for repeating experiments accurately. Conversely, when provenance can establish the quality and reliability of data, it prevents time-consuming and resource-intensive processing expended in recreating data.

The structure and content of a record of provenance can be complex because data, particularly derived data, often originates from multiple sources, multi-staged processing, and multiple analysis and interpretation. It may reference other data sourced from information and knowledge in a publication, output from an instrument data source, output from an application program employed in an *in silico* experiment, or a result set or derived data from another data resource. For example, an engine fault diagnosis may be based on; technical information from a component specification document, predicted failure data from a simulation run from a modelling application, a correlation identified from data mining a data warehouse of historic engine performance, and an engineer's notes made when inspecting a faulty engine component. In this example, the record of processing is based on; the simulation run modelled by an application; the retrieval by data mining of result data that matched the failure criteria predicted in the simulations; each step of statistical analysis, correlation, and visualisation carried out in analysing and interpreting the fault; and evidential reasoning from the engineer's notes.

Metadata provides the means to create a record of provenance in a consistent and structured way. Technical metadata specifies the name and physical location of a data resource or application program, or a publication reference. Contextual metadata specifies the integrity of a data resource in terms of conformance to naming conventions and structural relationships. Versioning specifies the currency of a data resource, an instrument configuration, or an *in silico* application. Together, contextual metadata and versioning describe the elements of the quality and reliability of data resources. Ownership is an indication of the quality and reliability of a data resource, particularly in the case of derived data based on subjective interpretations.

Data processing is rarely completed in single operation. Commonly, it involves multiple steps that represent iterative cycles of calibration, correction,

Document Control

transformation, merging, and aggregation through intermediate states. An instance of technical metadata specifies the name and physical location of an application that performs a processing step. One or more associated instances of technical metadata specify reference data and control parameters used in each step. For example, a cartographic application used to plot accurate maps would have associated technical metadata for the map projection constants, the area and scale of the map, and for digitised cartographic data. This is in addition to provenance information on the subject data shown on the map.

Several types of additional information are required to qualify the quality and reliability of data, particularly when data are derived from subjective interpretation. Annotation adds context to data through describing features, properties, and behaviour in the data, e.g. an annotation of a feature in a gene sequence. Descriptions of interpretations that explain how inferences and conclusions have been reached also improve the quality and reliability of data, e.g. a reference to an astronomical event in an historical chronicle may support the identification of an anomaly in a survey. A description can specify a data relationship or a correlation between data. It can also detail evidential reasoning to support an inference or conclusion. Evidential reasoning can reference a publication, or it can be based on knowledge, prior experience, or intuition, e.g. a reference to the same symptom being observed in other species.

It follows that a complete record of provenance can be lengthy, and may be represented in a complex data structures. This is particularly the case when other pieces of data, each with their own provenance, are involved in processing and deriving data, and when the granularity of provenance is not applied consistently across the processing steps.

The Grid must provide the capability to record a complete record of data provenance, and mechanisms for capturing provenance should be automated as far as possible to minimise need for manual entry. This implies that new Grid applications are built to exploit this capability. The Grid should provide tools to assist owners of existing data to create important provenance elements with the minimum of effort. It should also provide tools to analyse provenance and report on inconsistencies and deficiencies in the provenance record.

DATA ACCESS CONTROL

This section describes the requirements for controlling access to data in a Grid environment. It also describes the role access control plays in other data operations.

One of the principal aims of the Grid is to make data more accessible. Whilst there is no restriction over who can read data placed in the public domain, every science community and discipline has the need to control access over some of its data. This facility is required to ensure the confidentiality of the data is maintained. It is also required to prevent users who do have access to the data from making unauthorised versions of the data, or from changing its content in any way.

Control over access to data is managed by the owner granting and revoking privileges to others. In the Grid, it must be possible for the data owner to delegate authority to control access to one or more trusted third parties or custodians. This is a common requirement for data owned or curated by an organisation, e.g. Gene sequences, chemical structures, and many types of survey data.

Three main types of restrictions can be imposed over access to data. Restrictions can apply to data content, to the types of access operations that can be performed on content, and to the users. The facilities that the Grid provides to control access must be very flexible in terms of the combinations of restrictions and the level of granularity that can be specified.

The range in granularity that is required when specifying access to data content ranges from an entire database, or group of named data resources, down to the smallest horizontal or vertical subset of the contents of a single data resource. For example, the data available to astrophysicists may be limited to surveys owned by their own country. The granularity can apply to all existing versions, to a single named version of the data, or to data with a particular status. It must be possible to limit access to a subset of data by specifying the names of data items in the content and even the values they must match for them to be accessible. For example, in a clinical study it must be possible to limit access to patients' treatment records based on diagnosis and age range. It must also be possible to see the age and sex of the status of the patients without knowing their names, or the name of their doctor. The specification of this type of restriction is very similar to specifying data retrieval search criteria and matching rules.

Four types of privileges to access data are possible, read, insert, update, and delete privileges. By default, read privilege is always assigned to a user when access to any data is granted. The Grid must provide the ability to assign any combination of insert, update, and delete privileges to the same level of granularity to which read privilege has been granted. For example, an owner may grant insert access to every collaborator in a team so they can add new data to a shared resource. However, only the team leader may be granted privilege to update or delete data, or to create a new version of the data for release into the public domain.

Grid must provide the ability to grant access privileges to groups of users as well as to individuals. This facility is probably a necessity given the target population of Grid users worldwide, and the potential number of data resources available in a Grid environment. The grouping of users can be owner defined or community defined, and it may be hierarchical to reflect centre-based, national, and international communities and research teams. It may also be based on role or function. For example, read access may be granted to everyone with a role called research assistant or supervisor, or to everyone who is a biochemist or geneticist. Within the Grid it must be possible to apply the granularity of user-based privileges in conjunction with the granularity of data name and content privilege.

For access control to be effective it must be possible to grant and revoke all types of privileges dynamically. It must also be possible to schedule the granting and revoking of privileges to some point in the future, and to impose a time constraint, e.g. an expiry time or date, or a access for a specified period of time.

It is desirable that the Grid provides access control facilities that are easy to use, and that the facilities minimise the amount time owners and custodians need to spend in managing access privileges. Data owners in particular will be reluctant to grant privileges to others if the access control process is complicated, time consuming, or burdensome. The Grid must provide facilities that, whenever possible, enable access privileges to be granted to user groups declaratively. It must also provide tools that enable owners to review and manage privileges easily, without needing to understand or enter the syntax of the access control specification.

The degree of control that can be ultimately be achieved in restricting access to data will largely be determined by the internal structure and content of the data, and by the grouping of users. The more complicated the internal structure and the greater the degree of heterogeneity in data content, the greater the level of granularity that will be required in achieving fine-grained control over access. This is equally true for classifications and hierarchies in users groupings and roles.

The problem of complexity in data structure cannot be easily solved for existing data, but it can be minimised for data created in new Grid applications. It places an onus on researchers to consider access control requirements when they define project team and collaboration structures, and when they define research roles. It also places an onus on application developers to consider the access needs of the wider community, including other science disciplines, when they design data structures and when they define access control strategies for data.

DATA PUBLISHING AND DISCOVERY

This section describes the requirements for publishing and discovering data in a Grid environment. It also identifies the related operations involved in establishing the reliability and quality of data in the discovery process.

A principal aim of the Grid is to enable an eScience environment that promotes and facilitates sharing and collaboration of resources. There are a number of projects currently in definition or development, e.g. AstroGrid, that are intended to address this need by providing data curation and data publishing facilities. However, it remains a major frustration to scientists in all disciplines that they are not always aware of data in the public domain, and that they cannot always easily access the data they do manage to discover.

As an example, it is expected that 10,000 high energy physicists will access the centrally managed read-only data produced by the LHC experiment in CERN. At the other end of the scale there are an estimated 250,000 scientists worldwide carrying out, or with an interest in, genomic and proteomic research. They access data held in curated, standardised, reference databases. However, the great majority of bioinformatics scientists also create and maintain data locally as part of their own research. The Grid will not be successful if it does not succeed in making this and similarly held data more accessible to anyone who needs to use it.

The information that describes a data resource adequately when it is published is the same information needed to understand the context and usability of data resource when it is discovered. The requirements for this information, termed metadata and provenance, are specified in more detail other sections of this report. This section only describes the context of the information requirements in terms of the functional requirements for data publishing and discovery.

Information Requirements

The minimum information about a data resource that must be specified when it is published is a specification of its name, physical location, and ownership. A specification of the internal data structure is required for its content to be easily accessible. A specification of the characteristics of the data is required for a user to establish access permissions, and to make a judgement on the most efficient method of accessing the data. In addition, a specification of its logical structure and context, currency, ownership, and provenance is required to enable a user to establish the quality and reliability of the data content and so make a judgement on its value and use.

A major challenge to making data more accessible to other users is the lack of agreed standards for specifying and describing published data. There is an equivalent lack of standardisation in the procedures for publishing and discovering data. This problem is widespread, even in those disciplines where the centralised management and curation of data are well developed.

Document Control

It is important that the facilities the Grid provides for publishing data are extremely flexible. It is essential that data can be published in user defined structures, specifications, and descriptions. It is important that the publishing process does not necessitate moving the data to another site. It is also important that user are not forced to cede ownership or control over granting access rights to data they publish. Whilst it is desirable for the Grid to encourage standardisation, enforcing conformance to imposed standards must not be a pre-requisite for publishing data. Failure to achieve the required level of flexibility will be a major disincentive to users, and will result in much existing data and data created in the future remaining unpublished.

The Grid must support the ability to publish all types of data, regardless of size, and in any structure and format. This includes all numeric and mixed numeric and text data, documents and technical papers, images, and all data held in multi-media formats.

In some instances owners may need to supplement publication specifications with additional information that will improve the ability of a potential user to make more informed decisions about the value of data content. This form of annotation is useful when the content cannot be easily browsed on line because of its size, or because special tools are required to access or visualise the data. It may provide caveats or hints and tips in accessing and using the content. It may indicate when the content will become obsolete, when it will be superseded by a newer version. This type of annotation is also useful for describing data when restricted access rights have been granted to the data, or when only summaries or subsets of the entire resource are publicly available. It can provide information about the complete data resource, or can indicate when the content will move fully into the public domain. The latter is a specific requirement of the Astrophysics community.

Data Publishing Functionality

Functionality is required to support each step in the data publishing process. The steps are defining the publication specification, registering the data in a Grid environment, and deregistering data from a Grid environment.

Much of the functionality required for defining the publication specification is common with that required for defining and maintaining metadata. The Grid must provide the ability to enter and maintain specifications online manually. It should provide an online browsing capability for locating existing definitions in metadata catalogues, and the ability to reference an identical definition in publication specification. It should also provide the ability to modify the reference by redirecting it to a different version of the definition, or to a totally different definition. When no identical metadata definition is available, the ability should exist to copy the closest definition into the publication specification and then modify it by editing. It must also be possible to add additional descriptive annotation.

The Grid should provide the ability to register and deregister data resources dynamically within a Grid environment. It should be possible to schedule when these instructions are actioned, and to propagate them to sites holding replicates and copies of the resources. It should also be possible ensure the instructions are carried out when they are sent to sites that are temporarily unavailable.

It is anticipated that few owners will be prepared to make existing data more accessible if the publication process is onerous. Every opportunity in meeting the requirements must be taken to ensure that, wherever possible, the metadata definition publication specification processes are automated and that the burden of manual metadata entry and editing is minimised. There is a need for a set of intelligent tools that can process existing data by interpreting structure and content, extracting relevant metadata information, and populating definitions automatically. In addition, there is need for Grid applications to incorporate these tools into every functional component that interacts with any stage of data lifecycle so that metadata information can be captured automatically.

Data Discovery Functionality

The data discovery process can be broken down into three stages, data searching, data examination, and data evaluation.

The functionality required to support data searching needs to be as flexible as that required for data publishing. The minimum required to implement browsing and searching is that provided by an Internet browser and a search engine, e.g. Google.

The Grid needs to support two types of initial search capability, bounded and unbounded searching. In unbounded searching every known indexed catalogue in the Grid environment must be accessed in an attempt to satisfy the selection criteria. In bounded searching, the user must be able to specify which catalogues the search will be restricted to in order to satisfy the selection criteria. (note: an assumption is made that there will be one or more catalogues in the Grid environment to support data publishing and discovery). The user must also be able to frame selection criteria to establish the structure and relationships of target ontologies and naming classifications. In both types of search, the user must be able to specify the minimum selection criteria; typically these will be one or more key words, or a phrase of words containing operands.

When a search is successful, users must have the ability to drill down and review the structure, characteristics, and content of data resources returned in the results list. It must be possible to interrogate the structure and relationships within an ontology, to view the data in alternative ontology, and to review the data characteristics and additional descriptive information. It must also be possible to examine the contents of data resources by displaying samples, visualising, or statistically analysing a data sample or the entire data set.

In order to evaluate the quality and reliability of a data resource it must be possible to examine its currency and ownership. It must also be possible to navigate through its provenance, and the nested provenance of all data involved in its origin.

DATA RETRIEVAL

This section describes the requirements for retrieving data in a Grid environment. It also describes the role of metadata in specifying where data can be located, and in specifying the output content and format of retrieved data.

The ability to retrieve data within a Grid environment is a universal requirement. Users must be able to retrieve data directly into Grid applications or specialised tools when they need to process, analyse, annotate, and interpret data. They must also be able to retrieve data for review and examination during data discovery, and when performing data management operations.

All the participants in the analysis exercise expressed the need for a high degree of flexibility and control in specifying the target, the output, and the conditions of the retrieval.

Specifying the Target

It must be possible in the Grid to specify a single data resource, several combined or federated data resources, or a virtual database as the target for the retrieval. It should not be necessary to know the physical locations of any target specified, but users must be able to specify a preferred target when replicates and copies of data resources exist. However, the Grid needs the capability to override specified choices under certain conditions, e.g. to avoid failure when a specified target is unavailable, and when users specify performance or cost criteria on the retrieval.

Specifying the Output

It is important that users can specify the data content of result data without knowing the internal physical structures of any targeted data resource. When mappings exist, the Grid must provide the ability for users to specify result data items using metadata terms, and it must attempt to resolve any conflicts that arise when data referenced in one ontology is specified through another. Users must also have the ability to specify output formats that are different to those in which data are stored. This is particularly important when an application can only read data in one input format.

Specifying the Retrieval Conditions

The conditions for retrieving data are determined by search rules and data matching criteria. Some Grid users will have the ability to code these in specialised data query language and programming tools. However, the ability to code must not be a prerequisite for retrieving data in a Grid environment.

Users must be able to specify the target, output, and retrieval condition parameters using metadata, and the conditions for retrieval in semantics and syntax of the ontologies they understand. They must also have the ability to defer the retrieval to a future point in time, to specify preferred target locations when replicates or copies exist, to choose the fastest access path to the data, to specify the cheapest form of retrieval, and to abandon a retrieval after a specified processing period or elapsed time.

The requirements for data retrieval give rise to a number of implications for the capabilities that the Grid needs to provide.

First, the Grid must provide the ability to translate target, output, and retrieval condition parameters that are expressed in metadata terms into physically addressable data resources and data structures.

Second, the Grid must provide the ability to construct search rules and matching criteria in the semantics and syntax of query languages from the parameters that are specified, e.g. object database, relational database, semi-structured data and document query languages. It must also be capable of extracting data from user defined files and documents.

Third, when more than one data resource is specified, the Grid must provide the ability to link them together, even if they have different data structures, to produce a single logical target that gives consistent results.

Fourth, when linking data resources, the Grid must provide the ability to use data in one resource as the matching criteria or conditions for retrieving data from another resource, i.e. perform a sub-query. As an example, it should be possible to compare predicted gene sequences in a local database against those defined in a centralised curated repository.

Fifth, the Grid must be able to construct distributed queries when the target data resources are located at different sites, and must be able to support heterogeneous and federated queries when some data resources are accessed through different query languages. The integrated access potentially needs to support retrieval of textual, numeric, image data that match common search criteria and matching conditions. In certain instances, the Grid must have the ability to merge and aggregate data from different resources in order to return a single, logical set of result data. This process may involve temporary storage being allocated for the duration of the retrieval.

Sixth, when the metadata information is available and when additional conditions are specified, the Grid should have the ability to make decisions on the preferred time and preferred access paths to the data.

DATA ANALYSIS AND INTERPRETATION

This section describes the requirements for analysing and interpreting data in a Grid environment.

A principle aim of the Grid is to make data more accessible. This will provide a much greater opportunity for users to analyse and interpret data they have not created or do not own.

The purpose of analysis is to identify features, properties, and behaviours in data, and to identify correlations and anomalies between data. The purpose of interpretation is to explain what has been identified and to derive inferences and conclusions. Both activities in turn lead to the creation of data, information, and knowledge. The Grid must provide the ability to combine all data manipulation and data retrieval operations during the analysis and interpretation processes, and the ability to retrieve data directly into applications.

Within a Grid environment generalised tools may provide multidimensional and multivariate analysis capability. These tools are particularly important for mining historical data to identify trends, correlations, and anomalies, e.g. engineering data held in a data warehouse. Grid applications may be used to provide subject specific analysis techniques and methods. In both cases Grid must provide the ability to apply more than one analysis technique to the same or related data sets concurrently, and the ability to drill down to detailed data from summaries and aggregates.

The Grid must provide the ability to record inferences and conclusions drawn by assimilating evidence from each analysis and interpretation step in the process, and to capture the analysis workflow. The level of detail captured should be sufficient to represent an electronic lab book. It should also allow the workflow to be replayed in order to reproduce the analysis steps accurately and to demonstrate the provenance of any derived data.

When new data content is created, or existing data content is modified during analysis and interpretation, the Grid must be able to capture and save all the changes. It must also provide the ability to capture the related context, and where necessary create and modify associated metadata, e.g. by updating metadata mappings and by creating new versions of data resources.

There are a number of reasons why users may need to carry out analysis on locally maintained copies of data resources. It may be because interactive analysis would otherwise be precluded because network performance is poor, data access paths are slow, or because data resources at remote site have limited availability. It may be because the analysis is confidential, or it may be because security controls restrict access to remote sites. The Grid must have the capability to record when users signify that they have taken a local, or personal copy of data for analysis and interpretation, and must be able to alert users when the original data content changes. It must also provide facilities for users to consolidate changes made to a personal copy back into the original data. When this action is permitted, Grid must

Document Control

either resolve any data integrity conflicts automatically, or must alert the user and suspend the consolidation until the conflicts have been resolved manually.

METHODS OF WORKING WITH DATA

This section describes the standard methods of working with data in a Grid environment.

The requirements analysis identified two methods of working with data; the traditional approach based on batched work submitted for background processing, and interactive working.

Not surprisingly, batch working is the predominant method for operations that are compute intensive, process large volumes of data, and require limited user intervention. Typically, these types of operations can be scheduled to take place outside of peak processing periods, and when appropriate at remote sites.

Users tend to examine, analyse, and interpret the output of background processing interactively using tools that provide sophisticated visualisation techniques. However, the availability of affordable, scalable computing platforms is making it increasingly feasible to perform processing and other resource intensive operations, e.g. *in silico* experiments and simulations, online and to integrate them with interactive analysis and interpretation.

The Grid must provide the capability to maintain context throughout the workflow involved in batch and interactive working. This is necessary to ensure consistency in data content of the output of one workflow step that becomes input to the next. It is also necessary for reasons of provenance. The ability to maintain context is particularly important in an interactive method of working, when users can create context dynamically by integrating different types of data, and by running concurrent streams of analysis and interpretation. For example, a chemist creates context between different data types and representations when; an *in silico* experiment is captured on video; the experimental environment and observations on the molecule under investigation are recorded interactively in an electronic lab book; and when inferences and conclusions are similarly record. Equally, a geneticist creates context when dynamically defining the workflow in an interactive analysis session. The flow may involve; displaying different expressions of genes concurrently in conjunction with published literature sources and annotations from multiple sources; iteratively analysing the data to identify correlations and anomalies; and creating new annotations and recording conclusions.

The Grid must provide the ability to maintain context created between data of different types and representations drawn from different disciplines. It must also be able to maintain the context over a long period of time, e.g. the duration of a study. This is particularly important in interdisciplinary research, e.g. an ecological study investigating the impact of industrial pollution may create and maintain context between chemical, climatic, soil, species and sociological data.

DATA MANAGEMENT

This section indicates the likely orders of magnitude of data volumes in a Grid environment. It also describes the lifecycle of data, and the principal requirements for managing data.

Data Volumes

It is extremely difficult to predict the volume of data that could be held in Grid environment today, let alone in five years. However, a number of statements can be made about expected growth rates.

Technology trends, which in the areas of storage technology and networking are currently exceeding Moore's law, combined with regular price reductions are resulting in increasingly affordable scalable, high performance computing platforms with enormous online storage capacity. If it hasn't already happened, the time has almost arrived when it is cheaper to store data online than on tape archives, and it is certainly more reliable. In consequence, every science discipline is currently experiencing an explosion in data volumes, and annual growth rates in some disciplines that are almost exponential are not uncommon. The potential now exists to generate petabytes of data in a Grid environment.

The estimated total volume of data that will be retained annually from the LHC experiment currently in development at CERN is approximately twice the total amount of data currently held, and the experiment is planned to last for twenty years. By 2008, the planned LSST survey will be generating 5Pb of Astrophysical data per year.

Even in disciplines where data volumes until recently have remained relatively low in comparison with HEP, growth rates are increasing through the result of new analysis techniques being developed. For example, in Neuroscience, the ability to generate video sequence of brain sections has led to a marked increase in the content and volume of image databases.

Combinatorial Chemistry and Bioinformatics are two disciplines that have benefited from technology developments. The introduction of computing intensive *in silico* experimental techniques has led to some of the highest growth rates in data volumes. For example, it is estimated that a new gene sequence is generated every second, and that 16 billion bases had been sequenced by the end of 2001. Engineering is another discipline that has benefited in a similar way through the development of computing intensive simulation modelling techniques that generate large volumes of data for visualisation and analysis. Equally, cost effective storage has made it feasible to retain huge volumes of engineering design and performance data, and to subject historical data to mining analysis in order to identify trends and anomalies.

Digital libraries, the worldwide web, and more recently the emergence of XML as a standard text mark up language has led to an explosion in the volume of semi-

structured, searchable technical literature. For example, there are 150 million patent pages on molecule structures.

Data Lifecycle

The lifecycle of data consists of data creation, data modification, and data deletion. At any point in the lifecycle data can be referenced and used.

Data sources create raw data within a Grid environment. The value of output from an instrument or device tends to be limited until processing has corrected, calibrated, and transformed the data content into a more useable format and improved its quality. This is not necessarily the case for output from *in silico* experiments and modelling applications, which tends to be processed and output in a standard format.

Frequently, data processing involves a number of stages, some of which are iterative. The processing can be repeated on a periodic basis, e.g. raw event data at CERN are reprocessed annually. The need to reprocess data can also arise when processing parameters are refined, when the quality of the data is brought into question, and when it is necessary to repeat an experiment.

The output of each intermediate stage in processing can be considered to be read-only data. However, in the context of the data lifecycle, and in the context of provenance, it represents an incremental modification of the original raw data; which may involve data deletion in addition to data update. Dependent on the time and resources expended in each processing stage, and on the data volumes involved, it may be worthwhile to retain intermediate states for a fixed period, or until the next stage in processing has been validated. At minimum, the Grid must provide the ability to allocate temporary storage for intermediate states in data content, and the ability to secure them longer term online or offline. Grid must also provide the capability to capture each stage in modification to form a record of provenance.

Data analysis and interpretation employ iterative techniques. Data can be retrieved repeatedly, and examined repeatedly to describe and annotate the data. Numerical and statistical analysis techniques may be repeatedly applied to identify trends, correlations, and anomalies. These operations can result in entire copies or sub-sets of data being created. Annotation can result in derived data being created, repeatedly modified, and deleted. Interpretation can result in derived data being created, and in existing data being deleted, e.g. processed data proved to be incorrect, wrongly described annotations, and inferences and deductions that are no longer valid. Throughout this period of data volatility, the Grid must ensure that the integrity and consistency in data is maintained, that context is retained, and that necessary provenance information is captured.

Deletion represents the end of the data lifecycle. Data can be deleted at any point in the research process as illustrated above, but the principal reasons for data deletion is because the content is no longer valid, or because retention can no longer be justified. There is rarely a need to retain invalid data, although this is a requirement when

Document Control

intellectual property rights are involved, i.e. when knowledge has been patented. In this case, the Grid must provide the ability to retain such data online as a separate version with limited access, or in an offline archive that can be restored online when required.

The availability of affordable high capacity online and offline storage means that it is now common practice to archive data content at the end of its lifecycle, rather than deleting it as frequently happened in the past. The Grid must have the ability to distinguish between true online data and archived data. This is particularly important when archives are held online. It should be possible to discover the existence of archived data, but it must not be possible allow to retrieve archived data, or to amend its content in any way. The Grid should only ever allow two legitimate operations to be performed on an archive. It must be possible to restore the data online from archive, and to remove the archive; in which case the data will be lost forever.

Data Management Operations

The prospect of almost unlimited computing resources to create, process, and analyse almost unlimited volumes of data in a Grid 'on demand' environment presents a number of significant challenges. Not least is the challenge of managing effectively all data that can be discovered and accessed.

Given the current growth rate in data volumes, potentially billions of data resources of every type and size could be made available in a Grid environment over the next few years. The Grid must provide the capability to manage these data resources across multiple, heterogeneous environments globally, where required on a 24x7x52 hour availability basis. Data management facilities must ensure that data resource catalogues, or registries, are always available and that the definitions they contain are current, accurate, and consistent. This equally applies to the content of data resources that are logically grouped into virtual databases, or are replicated across remote sites. It may be necessary to replicate data resource catalogues, for performance or fail-over reasons. The facilities must include the ability to perform synchronisations dynamically or to schedule them, and they must be able to cope with failure in the network or failure at a remote site.

An increasing amount of the data in Science is being stored in complex data structures of mixed data format and representation. Complex data structures reflect rules and relationships in the data, and this in turn makes the need to maintain referential integrity more critical. A failure in referential integrity can result from media failure, from transaction failure, by failure in synchronisation, and by human error. Whatever the cause, it can potentially invalidate part or all of the data. Therefore, the Grid must provide the ability to recover the data to a consistent state at an earlier point in time. This requires that data resources are backed up regularly, and that when necessary recovery can take place online without loss of service.

An increasing amount of data held in complex data structures is volatile, and consequently the potential for loss of referential integrity through data corruption is

Document Control

significantly increased. The Grid must provide facilities that minimise the possibility of data corruption occurring. One obvious way is to enforce access controls stringently to prevent unauthorised users gaining access to data, either through poor security controls in the application or by any illegal means. A second, more relevant way, is for the Grid to provide a transaction capability that maintains referential integrity by coordinating operations and user concurrency in an orderly manner, as described in the related document, *Database Access and Integration Services on the Grid*, January 29th2002.

APPENDIX I INTERVIEW LIST & SOURCES

Interviews were held, questionnaires received from, and information supplied from by representatives of the following UK Grid and eScience related projects:

Project	Discipline	Principal Contact
AstroGrid	Astrophysics	Robert Mann, Edinburgh Royal Observatory
Comb-e-Chem	Chemistry and Bioinformatics	Jeremy Frey, Southampton University
Dame	Engineering	James Austin, York University
Discovery Net	Multidisciplinary – Bioinformatics, Environmental & Earth sciences	Yike Guo, Imperial College, London
Geodise (<i>information only from questionnaire</i>)	Engineering	Simon Kent, Southampton University
MyGrid	Multidisciplinary – Bioinformatics demonstrator	Carol Goble, Manchester University
Reality Grid (<i>Information only from presentation and published project scope and details</i>)	Multidisciplinary – Chemistry, Bioinformatics, Earth sciences	Peter Coveney, Queen Mary College, London

Further details of these projects can be found on: www.epsrc.ac.uk

Additional material was obtained from interviews and discussions held with the following Researchers:

Research Area	Research
Neuroscience	David Willshaw, Nigel Goddard, David Serratt, Douglas Armstrong, Edinburgh University
Ecology	Robert Muetzelfeldt, Edinburgh University
HEP	CERN users and Database support Group, contact Jamie Shiers
Astrophysics, AstroWise	Edwin Valentijn, Groningen Institute

APPENDIX II DATA REQUIREMENTS QUESTIONNAIRE

Grid Data and Information Needs Analysis

This template can either be used as a checklist when interviewing users, or as questionnaire.

For the purposes of this document the term data set is intended to refer to all: raw data from experiments and analysis; processed (corrected raw) data; reference data; inputs and outputs of models and simulations; and all information (contextual data) derived within the research process. The term data also includes numeric, textual, document, image and all other media forms.

When answering the template as a questionnaire, the responses should include what is currently done, and what is needed in the future. Please use **n/a** in the response box to indicated those questions that are not relevant to you.

Please note that despite the length of the template, this is not an exhaustive categorisation or an exhaustive list of questions. Please use the last section, **Additional Information**, to include questions and responses to any needs, problems, or issues not addressed in the current section. Please contact the author for any point of clarification.

Name	Email id	Telephone number	Group/Community Research Category
Research and Grid Interests			

Data Media and Volumes

Question	Guidelines	Response
On average how many data sets do you create and use annually?	Numeric, textual (include documents, image, etc) Give answers in orders of magnitude Q1-3	
What is the average volume of each type of data set?		
What is the average number of numeric and textual data items in each type of set?		
What is the average number of records in each type of data set?		
What percentage of data sets are created by you, by colleagues in your group, within your science community?		

Data management

Question	Guidelines	Response
What methods and techniques do you use to manage data sets?	Naming conventions, directory structures, urls, dbms, etc	
What structures do you use to store your data sets?	Files, Semi-structured data, databases (Object, Relational, Network, Hierarchical)	
What types of file structures do you use	ASCII, Binary, Indexed	
What is the average period of time your data sets are in normal use?	Days, weeks, months, years Project duration	
What percentage of data sets do you keep on-line (Disk)?		
What percentage of your data sets do		

Document Control

you secure off-line (Tape, CD Rom)?		
How often do you need to restore data sets on line?	Reasons, frequency	
Do you maintain personal data sets?	For periods of days/months/years	
What are the reasons for maintaining personal data sets?	Convenience, Reliability, Security Own data structures/applications	
What problems do maintaining personal data sets cause you?	Extra management effort Loss of currency	
What are the main deciding factors on how you currently manage your data?	Available tools and software Structure of published data Storage constraints	
To what extent is storage of the same data duplicated?	Give reasons, e.g. different formats, on/off line, speed of access, access control	

Data Discovery

Question	Guidelines	Response
Do you know where the data you are interested in is geographically located?	Provide country/centre details major locations and total number of locations	
Do you know who owns, manages, grants access to the data?	Same or different people?	
Do you easily find published data that you need to use?	Time spent searching Frequency of searching	
What methods and techniques do you use to search for published data?	Browsers & search engines Community web sites	
Is the data always in a standard/common format and structure	If not what format	
What additional information do you	Owner, origin, prior processing,	

Document Control

need on sourced data before it can be used?	age, currency, provenance	
What is the frequency of data downloads?	Give details of volumes and number of downloads by day, week, or month for types of data Specify number of down loads of same data and give reasons	
What problems do you encounter when you source data from others?	Access controls, ownership etc Can only see summary data	
What operations do you have to perform on the data before you can use it?	Validation, Conversion, Reformatting, Restructuring	
Are you aware of all the available sources of published data?		
What would make it easier for you to locate published data?		

Data Publishing

Question	Guidelines	Response
Do you or your group publish data for others to use?	Give volumes	
Is the data published in a standard/common format and structure	If not why not	
What operations do you perform on the data you publish it?	Validation, Conversion, Reformatting, Restructuring	
What additional information do you provide on data when it is published?	Owner, origin, prior processing, age, currency, full provenance	
What problems do you encounter when you publish?		

Document Control

Are you aware of who you're your data and how it is used?	How do you know	
What would make it easier for you to published data?		

Data Access Control and Sharing

Question	Guidelines	Response
In what ways do you limit access to you data sets?	Anyone can access Limited to named users, groups, specific roles Limited for a specified period	
Do you limit access to subsets of your data	Parts of data sets, specific data items	
Do limit what operations other can make on your data?	Insert, update, delete	
Do you log any/all access to your data?	How and why	
Do you overcome access control problems by providing copies of data sets or sub-sets?		
Do you always have access to the data you need to use?	When and why not	
What limitations do others impose on you when you access their data?	For what reasons	
How do you overcome problems in accessing other people's data?		
What are the most common access problems you encounter when others access your data?		
What are the most common access problems you encounter when you		

Document Control

access other's data?		
----------------------	--	--

Metadata

Question	Guidelines	Response
What technical metadata exists for the data you use in your work or do you create?	Physical location (path/url) Data structure and format Ownership and access permissions Date of creation/last update Versioning Data set Volume/size	
What contextual metadata exists for the data you use in your work or do you create?	Standard terminology, ontology etc	
Do you reference any/all of the data through contextual metadata?	What types of data and under what conditions	
Do multiple terminologies or ontologies exist for the data?	Which types and why	
What methods and techniques exist for mapping multiple definitions and resolving conflicts?	How do you resolve conflicts	
What provenance information exists for the data and what provenance do you need to know?	Creation source, processing, workflow routes, who interpreted the data	
Is provenance important in your work?	Reason	
How do you use provenance information?		

Objects (Information, methods, models)

Question	Guidelines	Response
Do you use object oriented techniques?	For what types of processing and what types of data	
Do you exploit object features?	Super classes, inheritance, abstract classes, polymorphism	
Do you make objects persistent?	What types of objects?	
Do you record provenance of objects?		
Do you store model definitions?	How	
Do you store model workflows?	How	
What problems do you commonly encounter when using objects?		

Versioning

Question	Guidelines	Response
Do you maintain versions of your data sets or objects?	Personal Working records for recovery	
At what level of granularity do you apply versioning to your data or objects?	Entire data set level One or more a subset levels	
Do you maintain versions across multiple data sets that are shared with others?	Work partitioned among several users, each with their own version	
Do you have to consolidate multi-data set versions to a base line?	Resolve all conflicts and differences between versions into a single validate version	
What methods and techniques do you use for version management?		
Do you only retain base-lined	For what reasons	

Document Control

versions, or all incremental versions?		
--	--	--

Data Access Methods

Question	Guidelines	Response
How do you access data	Scan, indexed, random	
Do you always access data through contextual metadata?	When and why Would you prefer to access data in this way?	
Do you access subsets of the data or filter data set retrieval	Subsets of complete records Subsets of data items for whole or part of data set Combinations of both	
What methods and techniques do you use to access/filter data	Selection criteria	
Do you parameterise data access?	Common access functions for which the retrieval is controlled by specified selection parameters	
Do you need to use parameters in one data set to filter access to another data set?		
Do you need to reproduce sub-set or filtered results?	Save the access query with parameters?	
What are the most common problems you encounter when accessing data?		

Data Operations

Question	Guidelines	Response
What reformatting and restructuring	Application specific formats	

Document Control

do you perform on data you retrieve		
What transformations do you need to perform on data you access?	Units of measure Calibration	
Do you need to merge multiple data sets or sub-sets before you can use it?	Create and keep or create dynamically when needed	
Do you produce summaries, aggregates or statistical profiles on data and save them as new data sets?	Purpose served	
Do you repeat the operations each time you retrieve the data or each time the data set changes?		
Do you require to 'drill down' into the original data sets from summary data you have created		
Do you insert new, or update or delete existing data in data sets?		
Do create linkages and maintain between data sets and versions	Include linkages between numeric, textual and rich media data forms	
Do you annotate data or add commentary to existing data or data you create	Thoughts, ideas, interpretations, findings, explanations	
Do you analyse multivariate data and store the results as new data sets	Multi-dimensional cubes of data What tools are used How do you store results	
What use do you make of remote computational resources?	Type of operations Method of data shipping Issues relating to accessing remote resources	
What are the most common problems you encounter when performing operations on data?		

Additional Questions

Question	Guidelines	Response

APPENDIX III COLLATED REQUIREMENTS AND DEPENDENCIES

Grouping	Requirement	Reason identified in analysis exercise	Dependencies
Data	Store numeric data to highest precision	Required for HEP, Engineering, Astronomy	
	Multilanguage support for text	Native language scientific journals	Multi-language support in the Grid
	Data annotated in mark up languages	Web pages, semi-structured data	XML
	Data in multi-media formats	Images from surveys Video records of experiments Electronic lab books	
	Data in relational and object DBMSs, and in XML	Existing data held in DBMSs, e.g. clinical, bioinformatics, chemical, ecological environmental, engineering, astrophysical, HEP	DBMS, XML, desktop application package connectivity and interoperability in the Grid
		Define complex models with hierarchical and network associations, ontologies, interdisciplinary research models, declarative simulation models	
	Data in spreadsheet formats	Personalised desktop databases created by individuals, e.g. gene and protein sequences, chemical structures	
	Modify data content – insert, update, delete	Data processing: calibration, correction, filtering and transformation Create and maintain: metadata, annotations, inferences (trends, correlations & anomalies) drawn during analysis and interpretation Correct or remove invalid data	Transactional capability to guarantee integrity of data
	Create and maintain context between data	Record naturally occurring and user defined relationships between data, e.g. provenance, interdisciplinary research	
	Create context between research data & published literature	Provide users with ability to access data referenced in a technical paper	

Document Control

Metadata	Create and maintain 'data about data'	Describe data characteristics	Agreed standards for describing data
	Technical metadata	Location, structure, characteristics required in data publishing & discovery, access & retrieval, and data management	
	Contextual metadata	Reference unambiguously data using accepted terminologies and ontologies within a science discipline Abstract referencing from internal data structures Improve quality of data by defining within rules of classification or ontology	Tools to create and maintain metadata Functionality to capture metadata from existing data automatically
	Ownership	Establish intellectual property rights, e.g. patented knowledge Credit owner when using data Seek access permissions Establish or resolve issues over quality & reliability of data	
	Mappings between metadata definitions	Compare ontologies and classifications, e.g. clinical, bioinformatics, chemical, neuroscience	
		Reference data through a preferred ontology, e.g. international medical or clinical records	
	Create and maintain rules	Inferred: required to resolve conflicts in mapping, express data value rules	
	Tools to create and maintain metadata	Inferred: from feedback, automated tools will encourage owners to populate metadata required for publishing	
Versioning	Maintain a history of changes to data, within a database, at the database level	Recover data to a previous state, e.g. to establish intellectual property status, recover lost or corrupt data	Agreed standards for versioning data

Versioning	Maintain multiple states of data online	Concurrent design and build, e.g. engineering, simulation models, software Collaborative research, e.g. personalised versions of data annotations and interpretations Multiple mappings between metadata and data, e.g. ontologies, classifications, coding systems Maintain context between cycles of raw data, processed data, and derived data Inferred: allow data to be copied or replicated before it is made available to users	Tools for creating and maintain versions
Provenance	Record the origin and derivation of data in terms of data involved, processing, and workflow	Establish the quality and reliability of data when published (databases, literature)	Metadata
	Review and interrogate the provenance of data	Accurately recreate data or experiment Establish quality and reliability of data discovered in order to assess its value and use; universal requirement	Tools to interrogate and examine provenance records
	Tools to create provenance automatically	Capture essential information in an electronic lab book, e.g. bionformatics & combinatorial chemistry Inferred: Avoid every Grid application building own functionality, enforce minimum standards	Functionality to capture provenance automatically
Access Control	Make data and literature available in the public domain	Universal requirement	Functionality to create and maintain access controls models
	Restrict public access to a subset of a database	Allow anyone to access summary and descriptive data, but not detailed data, e.g. astrophysical surveys	
	Grant access to named individuals, or to groups of users based on community, team, and/or role status, to any level of granularity	Enable groups to have access to unpublished data; e.g. supervisor can see researcher's data	
	Grant access to a subset of a database in terms of a specified set of data items within logical records that satisfy specified matching criteria, to any level of granularity	Most complex requirement in clinical research, e.g. research team can retrieve details and medical history of patients with a particular condition, but unable to identify patients or their doctor	
	Grant privileges to modify data content	Collaborative research in many disciplines	

Document Control

Access Control	Define and maintain an access control model based on groupings of users and privileges	Inferred: from interviews, make access control declarative	
	Delegate or assign granting authority to a third party or custodian	Curated data repositories	
Data Publishing	Register data in a Grid environment	Increase availability of data in a Grid environment, all disciplines Data and models referenced in published literature	Metadata Registers and catalogues to hold published specifications Tools for publishing data
	Remove registered data	Inferred: remove data from public or privilege access for any reason	
Data Discovery	Search and browse data registered in a Grid environment	Discover data, information, and knowledge in own or related discipline that is relevant to research; e.g. develop hypothesis, corroborate inferences and conclusions, provide input to simulation, analysis & interpretation	Data Publishing Tools to browse and examine discovered data content
	Examine and interrogate discovered data	Interrogate ontologies and classifications to understand semantics and syntax Establish quality and reliability of data to evaluate suitability for use	
Data Retrieval	Retrieve an entire database(s) or database(s) subset to produce a local copy	Static data for reference; e.g. read-only data relevant to subject of research Overcome network bandwidth or remote processing constraints on interactive working Confidential analysis and interpretation, e.g. bioinformatics research by Pharma companies	Metadata Connectivity and interoperability with all types of data resources and databases
	Retrieve a subset of database(s) that satisfies specified output format and data matching criteria	Retrieve data directly into a Grid application Limit scope of interest in data Take a statistical sample of a large database for initial analysis	Capability for the Grid to generate queries in language and DBMS specific semantics and syntax
	Retrieve a subset of database(s) that matches criteria specified in user held data	Drive the retrieval through a join or sub-query using own data, e.g. validate own gene sequence predictions against a curated repository	

Document Control

<i>Data Retrieval</i>	Retrieve a statistical summary or aggregate of data in a database(s)	Investigate characteristics of data content Search for trends, correlations, and anomalies in data	Abstracted data services
	Limit resources expended in a retrieval	Prevent long running queries that are resource costly, and degrade service levels for other users, e.g. astrophysics applications	Data characteristic to optimise queries
	Provide parameterised 'canned' queries	Constrain scope of data retrieval, most disciplines, or enforce use of standard access paths Provide readymade queries to users, most disciplines	Allocation of temporary storage to stage, transform, summarise, and aggregate result data
	Merge and filter data from multiple database sources	Federated and distributed queries in a heterogeneous environment, e.g. semantic searching, e.g. interdisciplinary research, browsing and searching	Scheduling capability
	Perform transformations on data during retrieval	Convert from one unit of measure to another, or one coordinate system to another, e.g. astrophysics data	
	Defer execution of retrieval	Schedule retrieval to take advantage of cheaper costs or availability of resources, or schedule workload, many disciplines	
	Specify data retrieval from preferred location	Inferred: Overcome network bandwidth constraints, or take advantage of remote site service availability	
Data Analysis & Interpretation	Drill down to detail data from visualisations, and summaries & aggregates	Validate trends, correlations, and anomalies identified during analysis, all disciplines	Ability to create and maintain context
	Personalise data	Describe all disciplines	Transactional capability
	Record inferences and conclusions	Explain properties, features, and behaviours in data, and explain trends, correlations, and anomalies	
Data Management	Copy and replicate data at a site, and across multiple sites	Ensure quality and availability of service, e.g. performance, availability, and fail over	Metadata, including versioning
		Curated repositories, e.g. bioinformatics, chemical, astrophysics, HEP	Resource synchronisation and coordination
	Synchronise copying and replication of data	Maintain data consistency	

<i>Data Management</i>	Change status of data and databases	Inferred: required to enable data to be made copied/replicated, or published before it is made generally available	Ability to exploit data management facilities provided by DBMSs, to ensure consistency in behaviour of applications that access existing data held in DBMS
	Maintain referential integrity of data in a database	Avoid data corruption and inconsistency, particularly in complex data models	
	Maintain transactional integrity of data	Avoid data corruption, inconsistency, loss	
		Transactions can be distributed	
	Backup and recover data	Ensure data can be recovered to a known state after data loss, e.g. through media failure, or corruption	
	Archive and restore data	Place cold data offline, and recover space Restore data when it is required	
	Consolidate multiple or personalised data versions into a single version	Inferred: For concurrent engineering, collaborative research, curated repositories	

APPENDIX IV REQUIREMENTS PRIORITISATION

The analysis exercise scoped a wide range of generic data requirements. A number of those identified are partially met within the existing Grid infrastructure. However, the participants in the exercise considered that none of current implementations met their needs fully. In particular, the lack of support for database management systems and semi-structured data appears to be a limiting factor on the extent to which Grid facilities can be exploited in projects.

A significant programme of work is required to implement all the requirements that have been identified, and it is not realistic to prioritise them all in this report. However, the following guidelines are intended to help in defining a schedule for the work. They are based on priorities expressed by participants in the analysis exercise, and on known dependencies within the requirements.

The highest priority identified is for the Grid to provide connectivity and interoperability with database management systems and XML-based data structures. This is not only important for the current UK eScience projects, but for all projects that are currently using these technologies to define complex data models. It is also important because it will increase accessibility to all existing data held in these systems and structure.

The ability to discover existing data is a high priority. This is dependent on the ability to publish data in registers and catalogues, and to describe published data in a meaningful way so that content can be easily accessed. The minimum that is required to provide this facility, and highest priority, is the ability to define the technical metadata elements of metadata in a register, particularly for data structures that are currently unsupported. Once this is available, the ability to define contextual metadata becomes a high priority, as this facility is required to abstract data services. Equally, there is a high priority to support versioning of metadata definitions.

The ability to retrieve data is a high priority once it becomes possible to publish and discover data. Data retrieval is necessary to examine data and to establish the provenance of published data. Initially, this requirement can be met by providing 'canned queries' for published data and its provenance, but it must soon be followed by distributed and federated queries. However, these will only be reliable when there is good quality technical and metadata available for published data, and the ability to perform both logical to logical, and logical to physical mappings. This will be dependent on the availability of metadata tools and automated functionality. Eventually, the Grid must be able to generate that language specific queries to examine data, interrogate ontologies, establish provenance, and to retrieve data into application. It is anticipated that all these facilities will be implemented incrementally.

It is a priority for applications currently under development to enable data content to be modified as well as created and deleted. This is particularly important in applications that provide analysis and interpretation functionality. Initially, the Grid should provide insert, update, and delete within a single database. Again, the ability

Document Control

to version data will be required to support this. It is anticipate that the facilities will be implemented incrementally, eventually providing distributed transactions.