e-Science Gap Analysis

30 June 2003
Geoffrey Fox, Indiana University
David Walker, Cardiff University

Part I: Introduction and Summary 6

1 Executive Summary 6

2 Introduction 7

2.1 Charge 7
2.2 Process 7
2.3 Overview of Report 7
2.4 Findings 9
2.4.1 Projects Surveyed (sections 3 and 4) 9
2.4.2 Grid Technology (section 5) 10
2.4.3 Grid Functions and Styles (section 6) 10
2.4.4 Gaps in Grid Styles and Functionalities (section 8.1) 10
2.4.5 Gaps in e-Science Runtime and Hosting Environment (section 8.2) 11
2.4.6 Gaps in Security (section 8.3) 11
2.4.7 Gaps in Workflow (section 8.4) 11
2.4.8 Gaps in Notification Service (section 8.5) 11
2.4.9 Gaps in Meta-data and Semantic Grid (section 8.6) 11
2.4.10 Gaps in Information Grid Technology (section 8.7) 11
2.4.11 Gaps in Compute/File Grids (section 8.8) 11
2.4.12 Gaps in other Grid Technology (section 8.9) 12
2.4.13 Gaps in Portals and Problem Solving Environments (section 8.10) 12
2.4.14 Gaps in Grid-Network Interface (section 8.11) 12
2.4.15 Education and Support Gaps (section 9) 12
2.4.16 Research Gaps (section 10) 12
2.4.17 Perception and Organizational Issues and Gaps (section 11) 12
2.5 Summary of Discussion in Sections 7, 8 and Appendix 13
2.5.1 Topics 1-6: Grid Types to Meta-data Services 13
2.5.2 Topics 7 to 11: Information Grid Services to Network Services 14

Part II: e-Science and its Technologies 15

3 e-Science 15

3.1 Particle Physics EDG and GridPP 15
3.2 AstroGrid 16
3.3 NERC DataGrid 16
3.4 DiscoveryNet 16
3.5 myGrid 17
3.6 Comb-e-Chem 17
3.7 RealityGrid 17

4 e-Business, e-Government, and e-Services 17

4.1 Campus Grids 18
4.2 Military and defence grids 18
4.3 Capital Radio Grid 19
4.4 Bioinformatics Grids 19

4.5 Distributed Aircraft Maintenance Environment (DAME) 20
4.6 Grid Enabled Optimisation and Design Search for Engineering (Geodise) 20

5 Grid and Cyberinfrastructure Technologies 21

5.1 Basic architecture Principles 21
5.2 Meta-data Rich Web Services Communicating by Messages 22
5.3 Level 1: Hosting Environments 22
5.4 OGSA and Level 2: OGSI Open Grid Service Infrastructure 23
5.5 Level 3: Permeating Principle and Policies 23
5.6 Levels 4, 5, and 6: System and Application Grid services 25
5.7 Grid Technology Suites 25

6 Functionalities and Styles of Grids 26

6.1 Introduction 26
6.2 Compute/File Grids 27
6.3 Desktop Grids 27
6.4 Information Grids 28
6.5 Complexity Grids 29
6.6 Campus and Enterprise Grids 29
6.7 Grid Characteristics 30

7 Status of e-Science Service Development 31

7.1 Types of Grid 32
7.2 Core Infrastructure and Hosting Environment 33
7.3 Security Services 34
7.4 Workflow Services and Programming Model 35
7.5 Notification Services 39
7.6 Registry, Metadata and Information Services 40
7.6.1 Introduction 40
7.6.2 Basic metadata including Registry 41
7.6.3 Information Aggregation Web Services 42
7.6.4 Semantically rich Services and Ontologies 42
7.6.5 Provenance 43
7.7 Information Grid Services 43
7.8 Compute/File Grid Services 44
7.8.1 Planning and Management 45
7.8.2 Scheduling 46
7.8.3 Access to Remote Computers 46
7.8.4 Managing Job Submission 46
7.8.5 Compute/File Grid Shell: Data Transfer 46
7.8.6 File and Storage Access 46
7.8.7 Replica Management and Caching Technology 47
7.8.8 Virtual Data 47
7.8.9 Parallel Computing 47
7.8.10 Job Status 47
7.9 Other Grid Services 47
7.9.1 Grid Shell 47
7.9.2 Accounting and Grid Economies 47
7.9.3 Fabric Management 48
7.9.4 Visualization Datamining and Computational Steering 48
7.9.5 Collaboration 48
7.9.6 Packaging 49

7.9.7 Other Technologies 49
7.10 Portal Services and Problem Solving Environments 49
7.11 Network Services 51
7.11.1 Network Performance, Monitoring and Information Systems 51
7.11.2 Network Services and Operations 52
7.11.3 Network Reservation 52

Part III: Gap Analysis 53

8 Grid Technology Gaps 53

8.1 Different Types of Grids and their Federation 53
8.1.1 Introduction 53
8.1.2 Summary of Gaps 54
8.1.3 Jini 54
8.1.3.1 Grid Prototypes Based on Jini 54
8.1.3.2 Jini in Agent-Based Grids 55
8.1.4 JXTA 55
8.1.4.1 Critique of JXTA for Building Peer-to-Peer Grids 55
8.1.4.2 Use of JXTA in a Grid for Capital Radio Group 56
8.1.4.3 JXTA GAT Binding for the Triana Project 57
8.1.5 .NET 57
8.1.5.1 Overview of .NET 57
8.1.5.2 Grid Middleware Based on .NET 58
8.1.6 Grids and Virtual Private Networks 58
8.1.7 Grid Federation 59
8.1.8 Dynamic Configuration (Autonomic Computing) 60
8.2 e-Science Runtime and Hosting Environment 61
8.2.1 Introduction 61
8.2.2 Summary of Gaps 62
8.2.3 Selected Snapshots of Current Activities 62
8.2.3.1 Invocation Framework from IT Innovation 62
8.2.3.2 ICENI From Imperial College 63
8.2.3.3 Messaging Infrastructure 64
8.2.3.4 Lightweight Middleware 64
8.3 Security Infrastructure 66
8.3.1 Introduction 66
8.3.2 Summary of Gaps 66
8.3.3 Current Globus and related (EDG GriPhyn Condor) Technology 66
8.3.4 Possible Future Projects 67
8.3.4.1 Harden VOMS from EDG 67
8.4 Workflow 67
8.4.1 Introduction 67
8.4.1.1 Workflow representation and composition 68
8.4.1.2 Workflow enactment (Compilation and Execution) 68
8.4.2 Summary of Gaps 69
8.4.3 Selected Snapshots of Current Activities 69
8.4.3.1 Workflow at Southampton 69
8.4.3.2 Workflow at Newcastle 69
8.4.3.3 Workflow at Cardiff 69
8.4.3.4 Workflow at IT Innovation 70
8.4.4 Possible Future Projects 71
8.4.4.1 Newcastle Workflow Engine 71
8.5 Notification Service 71
8.5.1 Introduction 71
8.5.2 Summary of Gaps 72

8.5.3 Selected Snapshots of Current Activities 72
8.5.3.1 myGrid and Wrapped JMS (Java Message Service) at Southampton 72
8.5.3.2 Notification Grid Service at Newcastle 72
8.6 Meta-data and Semantic Grid 73
8.6.1 Introduction 73
8.6.2 Summary of Gaps 74
8.6.3 Globus Specific Gaps 74
8.6.4 Selected Snapshots of Current Activities 74
8.6.4.1 UDDI Evaluation 74
8.6.4.2 Unicore and MDS 75
8.6.4.3 Relational Grid Monitoring Architecture 76
8.6.4.4 Semantic Grid Work at Southampton 77
8.6.4.5 SDT Semantic Discovery Toolkit 77
8.6.4.6 Metadata Management 77
8.6.4.6.1 General Scientific Metadata Format 78
8.6.4.6.2 CCLRC Data Portal 78
8.6.5 Possible Future Projects 78
8.6.5.1 Semantic Grid Expectations 78
8.6.5.2 A Semantic Grid Service Registry from IT Innovation 79
8.6.5.3 Provenance Specification 79
8.6.5.4 SRB Evaluation at CCLRC e-Science Centre 79
8.7 Information Grid Technology including OGSA-DAI 80
8.7.1 Introduction 80
8.7.2 Summary of Gaps 81
8.7.3 Current OGSA-DAI 81
8.7.3.1 North-East e-Science Centre OGSA-DAI Activities 82
8.7.4 Selected Snapshots of Current Activities 83
8.7.4.1 Environmental Grids 83
8.7.4.2 AstroGrid 83
8.7.5 Possible Future Projects 84
8.7.5.1 Capital Radio 84
8.7.5.2 Futures of OGSA-DAI 85
8.7.5.3 Environmental DataGrid 85
8.7.5.4 AstroGrid 86
8.8 Compute/File Grids: Scheduling, Access to Mass Storage, Replica Management and Virtual Data 86
8.8.1 Introduction 86
8.8.2 Summary of Gaps 87
8.8.3 Possible Future Projects 88
8.8.3.1 Hardening of EDG Replica Technology 88
8.8.3.2 Hardening of EDG Compute Resource Broker 88
8.8.3.3 Hardening EDG Storage Element - Grid interface to mass storage resources 88
8.9 Other Technology areas 89
8.9.1 Introduction 89
8.9.2 Summary of Gaps 89
8.9.3 Selected Snapshots of Current Activities 90
8.9.3.1 Grid Economies and Markets 90
8.9.3.2 Improve collaboration (Access Grid) technology 91
8.9.4 Possible Future Projects 91
8.9.4.1 GridWeaver: an enhanced LCFG for e-Science wide fabric management 91
8.9.4.2 Development of a Gridmake 91
8.9.4.3 Debugging tools 92
8.9.4.4 Grid-Based Visualisation Services at Leeds 92
8.9.4.5 Visualization and Computational Steering at Cardiff 93

8.9.4.6 Grid Visualisation Services at CCLRC 94
8.9.4.6.1 GAPtk visualisation server 94
8.9.4.6.2 Application programming interfaces 95
8.9.4.6.3 Applications 95
8.10 Portals and Problem Solving Environments 96
8.10.1 Introduction 96
8.10.2 Summary of Gaps 96
8.11 Grid-Network Interface 97
8.11.1 Introduction 97
8.11.2 Summary of Gaps 97
8.11.3 Selected Snapshots of Current Activities in Monitoring and management 98
8.11.3.1 Network Measurement and Monitoring 98
8.11.3.2 Network Information Services 99
8.11.3.3 Network Performance Services 100
9 Education and Support Gaps 100
9.1 Gaps 100
10 Research Gaps affecting near-term e-Science (issues where not enough is known to allow "development" project) 101
10.1 Gaps 101
11 Perception and Organizational Issues and Gaps 101
12 Acknowledgments 103
Part IV: UK Open Middleware Infrastructure Institute (OMII) 104
13 Core e-Science Middleware Action Plan 104
13.1 Elements of the Action Plan 104
13.1.1 Grid Systems 104
13.1.2 Basic Technology 105
13.1.3 Essential Services 105
13.1.4 Core Domain Grid Services 105
13.1.5 Programming Environments 105
13.1.6 Portals and User Interfaces 105
13.1.7 Other Grid Services 105
13.2 Agile Development of Grid Middleware 106
Part V: Appendices 107
14 Detailed Descriptions of UK Grid Services and Activities 107
15 Worldwide Grid Resources 107
15.1 Glossary of Concepts 107
15.2 National and Regional Programmes 109
15.3 Descriptions of Projects 110
15.4 Descriptions of Systems and Tools 121
15.5 References 137

Part I: Introduction and Summary

1 Executive Summary

We present the results of personal interviews from mid February to early April 2003 of 80 scientists concerning the current state of Grid and Cyberinfrastructure technology with respect to their use in e-Science. The guiding theme of each interview was the identification of issues affecting the development by 2006 of robust e-Science infrastructure that would be generally usable by both large and small groups wishing to set up or join virtual organizations. Our coverage of the UK e-Science participants was reasonably thorough and we augmented this by discussions with some European and US Grid experts and, of course, attendance at meetings like GGF7. The study also included substantial material gathered both to cover specific topics for this report and to document current e-Science project progress with attention to Web and Grid services developed. The main body of this report is augmented by a public appendix listing the back up material with a tabular summary of the information gathered. Also included in the appendix is a glossary of concepts related to the Grid and e-Science, and brief summaries and URLs of projects, tools and infrastructure in these areas. The notes from the interviews have been kept confidential. The report has summarized the interviews as a set of over 120 comments (on issues or gaps) and used community understanding and these gaps to produce two taxonomies; one of Grid technologies and another of Grid functionalities. These taxonomies are consistent with OGSA – the Open Grid Service Architecture – of the Global Grid Forum. We have used them to classify issues directly connected to the technology and functionality of Grids. Some of the gaps were further classified under education and support, research and finally perception and organizational issues. The gaps, summaries and taxonomies can be read and interpreted by each reader of this report. The identification of gaps is further informed by a survey, presented in Section 7, of worldwide Grid developments relevant to e-Science, based on the literature and information on web sites.

However, we went further than just identifying the gaps, and developed a possible development program, presented in Section 13, which aims to deliver by 2006 Grid technologies supporting the e-Science functionalities identified in this report. This program is designed to support the full e-Science community including disciplines like particle physics needing highly robust large scale “Compute/File” Grids as well as the Information Grid based applications such as those in Bioinformatics and Virtual Observatories. Our plan will continue the UK leadership in areas like OGSA-DAIS and the Semantic Web; it covers Grids designed to support the activities of the current core e-Science projects, campus Grids and the military and commercial Grid areas we studied. Our software development program is structured as a central activity responsible for overall direction and the architecture and software engineering. Our report notes that many styles of Grids are needed with autonomic (fault-tolerant), lightweight and peer-to-peer characteristics being highlighted by our analysis. The central activity would be responsible for this core Grid infrastructure which could of course be developed with other groups in the UK or around the world. We identify the need for federation technology to allow Grids of different styles and suppliers to be integrated and suggest OGSA can be used for federation as well as interoperability. The central effort of the proposed software development program is accompanied by distributed activities. The total effort is divided into basic technology, essential services for any Grid, core Grid services for the two most important functions (Information and Compute/File), programming environments, portals and user interfaces, and the outer level discipline-specific services. Many of these services are under development in the current e-Science program but without central coordination and motivated largely by a particular application and near term demonstrations. We suggest that requirements and technical issues are well enough understood to allow, for example, the development of UK e-Science workflow, notification and information services – the capabilities identified under essential services. We emphasize the basic technology category including the so-called hosting and execution environments which includes asynchronous messaging, network monitoring, Grid federation, a security infrastructure supporting fine grain authorization, and a component model optimized for e-Science. Our study covered software engineering issues and this should be a major strength of the central coordinating activity.

Such a well-designed software development program would attract international attention and collaboration. It should be accompanied by a support and education program recognizing the inherent

problems in supporting rapidly varying technology and informing the community about both today's best practice and expected future directions of e-Science.

The appendix provides, in Section 14, a detailed description of UK activities related to e-Science, while Section 15 summarises useful information compiled in the course of researching this report, in the form of a glossary, and short descriptions and URLs of relevant Grid projects, tools, and systems.

2 Introduction

2.1 Charge

The authors were asked to produce a detailed 'Gap Analysis' report of present and foreseen Grid middleware with the main aim of identifying the areas in which the middleware is lacking or poorly specified, i.e., the gaps. The report should also:

1) Recommend actions to be taken to deliver an alpha release of a UK open source/open standard Grid middleware stack by Q4 2004 with a production version by Q1 2006.
2) Explicitly address the issues identified at the Technical Discussion Meeting on Grid Middleware in December 2002.
3) Summarise the middleware products of the UK Core e-Science Programme. This includes services, XML schema, and tools.
4) Include a discussion of software engineering, interoperability, and federation issues.

2.2 Process

This gap analysis report is based mainly on interviews with members of the UK e-Science community. In addition, several members of the community have contributed to the report by writing short sub-sections on their particular area of expertise – these are acknowledged in Section 12. A third source of input was documentation from some of the e-Science projects. The transcripts of the interviews remain confidential, but all other material appears in this report.

Interviews were conducted with representatives of the following:

1) The UK e-Science Centres and the Grid Support Centre.
2) All the EPSRC and PPARC pilot projects, as well as the NERC-funded GODIVA project, the OGSA-DAI project, and the European DataGrid project.
3) Companies involved in producing Grid middleware, such as IBM and Hewlett-Packard.
4) Companies and organisations that are potential users of Grid middleware, such as the European Bioinformatics Institute, CERN, and Capital Radio Group.
5) Individuals with expertise in particular areas related to Grid middleware, such as security, networking, and software engineering.

2.3 Overview of Report

In sections 3 and 4, we describe the applications looked at and summarize some key features that lead to the gaps described later in section 8. We first discuss e-Science where the coverage is reasonably complete among the major academic areas that can exploit this idea today. The next section describes the issues in business, government (military) and what we term e-Services; the latter is illustrated by campus Grid projects [WhiteRose] and the broad capabilities provided by the European Bioinformatics Institute [EBI] to support bioinformatics. The coverage of topics in these areas is not as complete as that in e-Science for in some sense e-Science is leading the development of Grid systems. Nevertheless it is very important to consider a broader base than e-Science as the latter must leverage commodity technologies and develop its infrastructure as much as possible in alignment with the much larger business and consumer markets. Sections 5 and 6 respectively give a broad overview of Grid technology [Foster99A] [Foster01A] [Foster03A] and functionality to act as a backdrop to the later discussions. This is augmented by the glossary of section 15.1. Section 7 summarises the current status of Grid computing from a world-wide perspective. It classifies the Grid services in the same 11 broad areas used in Sections 7 and 8 and depicted

in fig. 2.1 below. Section 2.5 has a table listing much of the material in sections 7, 8 and the Appendix (section 14) classified into the 11 areas we chose for organizing our material.

Sections 8 to 11 are devoted to the results of the gap analysis. The first describes the project management and software engineering issues that would be important if the UK (possibly in collaboration with a broader open middleware initiative) core e-Science activity mounted a major project to develop middleware. We note that for the purposes of this analysis, we define the software used for e-Science as Grid technology, whether it uses Globus, .NET, Jini or some other core infrastructure. Section 8 is the heart of the report and presents in 11 sub-sections those areas of Grid technology where gaps exist with enough understanding to warrant consideration in a middleware initiative. Figure 2.1 depicts Grid architecture at a high level in order to explain the different sub-sections of Section 8 which presents the gaps organized as explained in this figure. The gaps described in sub-section 8.x correspond to the world-wide service summary given in sub-section 7.x.

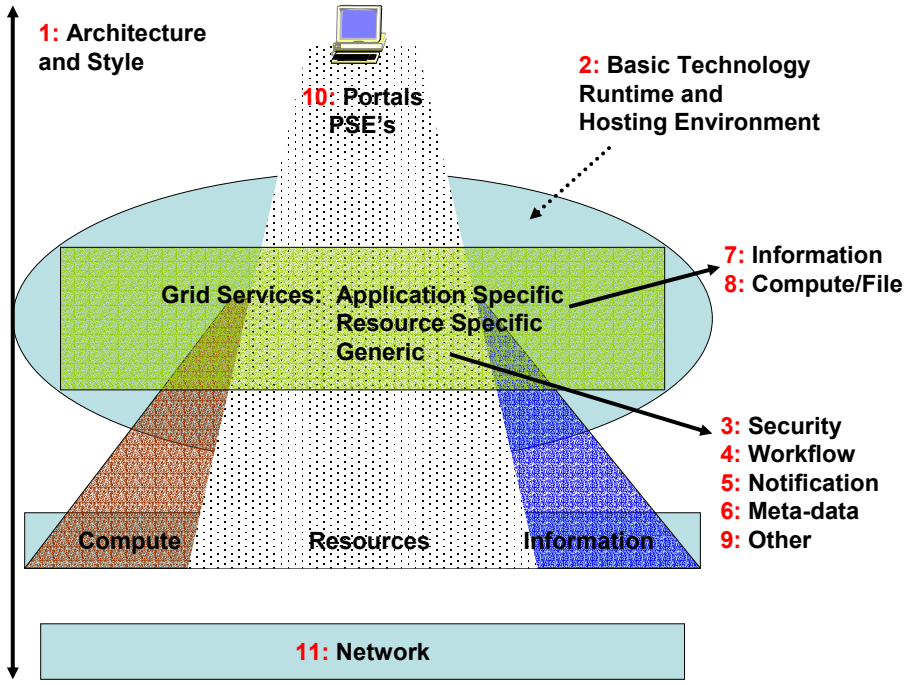


Fig. 2.1: Classification into 11 areas of the Grid middleware gaps identified

Section 8.1 notes that with our broad definition, there are many functions and styles of Grids differing in both size, heterogeneity, dynamics and needed functionality. This diversity is generated by different application requirements and different choice of base technologies. We describe the architectural implications of this and how it can be addressed by allowing multiple internal technologies and supporting a Grid formed by composition and federation of other Grids.

The middleware (services) must be supported by some run-time which can differ both between grids and in fact internally to a Grid. For example, if we link a PDA to a supercomputer in a Grid, these resources are likely to have different run-time environments even though the may connect through common protocols or API's. Section 8.2 discusses this important topic both in terms of specific capabilities and broad technology choices.

Sections 8.3 to 8.6 discuss four important services highlighted in OGSA and most distributed object systems like CORBA and Java RMI. These start with security where our treatment is incomplete, as there is a separate task force addressing this. We stress the analogies with Virtual Private Networks which are accepted practice and the value of a federated security architecture to provide the needed fine grain

authorization for Grid resources. Section 8.5 describes notification which is part of the new OGSI standard but there is no clear consensus yet on application requirements here. Section 8.4 describes the very important workflow service (or distributed application/service integration) where there is probably good agreement on functionality but less consensus on how to express it. Many e-Science projects require a powerful workflow engine capable of dealing with the high service to service bandwidth and dynamic nature required in e-Science. The middleware effort here involves both supporting a language to express workflow and the enactment engine to efficiently implement it. As in conventional programming where one has multiple languages often supported by a single runtime, we can expect a good workflow runtime (enactment) to support substantial experimentation with workflow languages. Section 8.6 describes a major area that many projects highlighted as critical; this is area of generating, storing and accessing meta-data; we define this broadly to span both the simple look-up of Web services as in UDDI to the richness of the Semantic Grid. The need for supporting semantic interoperation of services is a broad requirement although the design of meta-data/look-up services that span high and low levels of the Grid remains unclear.

Sections 8.7 and 8.8 focus on the capabilities needed to support information and compute/file grids respectively. Information grids have been a special focus of the UK e-Science program and building on the success of OGSA-DAI form an important set of activities where the requirements of the astronomy and bioinformatics communities are reasonably clear. Other application-specific services such as those in Earth and Environmental science can be expected to become clearer as projects in these areas mature. Section 8.8 is dominated by the needs of the particle physics community and here one finds the greatest overlap with existing European and US efforts. Campus (computing) Grids can also be expected to grow in importance. Replica management, access to storage, scheduling and virtual data are major compute/file Grid areas. The scale of the particle physics problem emphasizes the need for robust well-managed grids.

Section 8.9 is a gallimaufry of services that did not fit conveniently into the above categories; accounting, Grid fabric configuration and management, visualization, Grid economies and collaboration are discussed here.

Section 8.10 describes the technologies to support portals and problem solving environments. Here there are some emerging technologies that should allow greater sharing of portal and PSE components between different e-Science projects. This effort has been pursued by the GCE (Grid Computing Environments) working group of the GGF which has produced two “Grid Information” documents describing this area. We expect that the e-Science program could usefully adopt and extend some of these ideas.

Section 8.11 describes issues on the interface of Grids and the physical network. Grids require sophisticated monitoring, and good API’s to first query network performance and then reserve or predict network capacity. Further Grids demand end-to-end performance which requires coordination between backbone and regional network providers.

Sections 9 through 11 present less technical gaps starting with a discussion of education, training and support issues. The dynamic breadth of Grid technology is particularly challenging in this area. The research issues in section 9 could possibly be in section 8; they represent interesting areas where we felt that more research was needed before they could be part of the broad deployment. We emphasize that all areas of Grid and e-Science still need intense research and our division between sections 8 and 9 is rather arbitrary. Section 11 contains a set of interesting observations which represent the “confusion gap” caused by the many different views of the Grid and the rapidly changing state of its technology.

2.4 Findings

Here we summarize the major conclusions and features of the report labeled by particular sections

2.4.1 Projects Surveyed (sections 3 and 4)

We examined in detail several projects in detail including

- **Particle Physics:** EDG (European Data Grid), GridPP and LCG (LHC Computing Grid)
- **Bioinformatics:** DiscoveryNet, myGrid and the European Bioinformatics Institute EBI

- **Combinatorial Chemistry:** Comb-e-Chem supporting the electronic laboratory
- **Material Science:** RealityGrid supporting simulations and computational steering
- **AstroGrid:** The important "Virtual Observatory" class
- **Military and Defence:** very heterogeneous Grids from logistics to dynamic combat situations
- **Capital Radio Grid:** Peer-to-peer environment for distributing multimedia material
- **Real time diagnostic Grid:** DAME with major industry (Rolls Royce) participation
- **Engineering Design and Optimization Grid:** Geodise with major aerospace industries
- **Environmental Grids:** the NERC Data Grid and GODIVA for oceanography

2.4.2 Grid Technology (section 5)

We classify the building blocks of a Grid into six layers starting with

- **Level 1:** meta-data rich Web services with message-based input and output
- **Level2:** The Grid service component model defined in GGF standard OGSI (Open Grid Service Infrastructure)
- **Level 3:** A set of "Grid permeating principles and policies" that characterise the execution run-time or enhanced hosting environment
- **Level 4:** The key Grid functionalities that "all Grids" should have with (emerging) OGSA specified interfaces
- **Level 5:** Somewhat unclear distinction from level 4 of other system Grid services
- **Level 6:** Applications formulated as Grid services

We briefly mention particular Grid technology suites including the Globus toolkits (GT2 and GT3) and Unicore.

2.4.3 Grid Functions and Styles (section 6)

We analysed the functionality and style of operation of Grids and identified six important broad functionalities.

- **Compute/File:** Grids largely aimed at supporting typical scientific jobs such as those used in particle physics data analysis
- **Desktop:** The internet computing style of Grids as used by Entropia and United Devices. The climateprediction.net Grid uses this functionality.
- **Information:** This type of Grid involves integration of large scale distributed data repositories and is a great strength of the UK e-Science program.
- **Complexity:** This hybrid Grid links Information and Compute/File Grids and can be expected to be of growing importance.
- **Campus and Enterprise:** We expect growing interest in Grids to drive institutional computing/information technology facilities.

We also identified several styles of Grid operation including Semantic, Peer-to-peer, Lightweight, Collaboration and Autonomic Grids.

2.4.4 Gaps in Grid Styles and Functionalities (section 8.1)

- The architecture and technology base for a Grid determines some of the important styles discussed in section 6 and we identify Autonomic, Lightweight and peer-to-peer as important Grid characteristics. These are either three separate Grids or customizations of one or two basic technologies
- Campus and Complexity (hybrid) Grids need attention with initial efforts focusing on Information and Compute/File Grids
- Federation is a natural way of linking Grids of different styles and functionalities
- It is important to have very good project management and software engineering in any such major software initiative.

2.4.5 Gaps in e-Science Runtime and Hosting Environment (section 8.2)

- Although one should move to Grid Service (OGSI) standards, it is important to maintain concurrence with W3C and GGF standards which will hopefully converge. Approaches that hide W3C and GGF differences should be investigated
- One could design and develop a community wide "ScienceBean" which would be better suited than Enterprise Javabeans for large data transfer in e-Science
- A powerful messaging layer is needed to support federation and robust management tools
- One needs a scalable fault-tolerant management framework including support of notification and provenance
- Support of languages like Perl is critical for many existing applications such as Bioinformatics
- There is an opportunity to use modern middleware technologies

2.4.6 Gaps in Security (section 8.3)

- This area is addressed by a separate task force but the need for better certificate infrastructure and fine grain dynamic authorization were common themes
- We elaborate on the generalization of VPN's (Virtual Private Networks) to VPG (Virtual Private Grid)
- A Federated Security architecture can give fine grain authorization

2.4.7 Gaps in Workflow (section 8.4)

- Workflow or "orchestration/integration of Grid services" was emphasized and the UK e-Science has several activities which could together lead to development of a "production quality" community system
- The workflow system should support parallelism, separate runtime & language (so one can support multiple languages with same runtime) and ontologies describing Grid service semantics

2.4.8 Gaps in Notification Service (section 8.5)

- OGSI notification simplistic but it is part of any robust Grid infrastructure
- As with workflow, current e-Science activities could be "combined" to lead to a production quality" community system

2.4.9 Gaps in Meta-data and Semantic Grid (section 8.6)

- The need for meta-data enabling semantic interoperability of services was identified as a major gap
- The current tools (UDDI, MDS and the improved RGMA) need to be enhanced
- Tools from the Semantic Grid will be valuable in generating and reasoning about meta-data
- There are of many non-Grid metadata organizations whose repositories need to be wrapped
- The new Service Data Element model of OGSI suggesting metadata is stored in the applicable services needs investigation
- Provenance will be very important

2.4.10 Gaps in Information Grid Technology (section 8.7)

- The work of OGSA-DAIT including distributed query and extended database support builds on the widely recognized success of OGSA-DAI
- Applications like Bioinformatics and Astronomy need "filters" (transformations) between the user and databases. Further OGSA-DAIT will need to integrate with workflow activities
- Peer-to-peer style Information Grids are interesting
- Many existing (non OGSA-DAI) databases will need to be wrapped

2.4.11 Gaps in Compute/File Grids (section 8.8)

- This is the best developed area of Grids but robust infrastructure is essential
- Compute/File Grids are critical in particle physics and Campus Grids

2.4.12 Gaps in other Grid Technology (section 8.9)

Several Grid services came up that do not fit neatly into previous categories – these include collaboration, accounting, visualization, fabric management and a possible "gridmake"

2.4.13 Gaps in Portals and Problem Solving Environments (section 8.10)

- We suggest a more coordinated approach across projects could lead to greater re-use of portal components.
- The portlet architecture endorsed by the Grid Computing Environment (GCE) research group of the GGF is promising

2.4.14 Gaps in Grid-Network Interface (section 8.11)

One needs end-to-end monitoring of the network supporting both performance measurements and reservation. This capability is best integrated with the low-level Grid runtime environment

2.4.15 Education and Support Gaps (section 9)

Two highlights in the education and support area were need for testbeds and the difficulties of supporting rapidly changing technology

2.4.16 Research Gaps (section 10)

There is much research needed in all areas of the Grid. Highlights include service management, fault tolerance, and Grid adaptability. Grid debugging and performance (benchmarking) also needs attention.

2.4.17 Perception and Organizational Issues and Gaps (section 11)

- There is intrinsic conflict between "demonstrator" projects (which characterises many current successful activities) and producing generally useable software
- The Grid is too hard
- There is a lot of confusion as to current and near future state of the Grid; for example the difference between OGSI and OGSA is not clear. Decision-making would be helped if current status and future Grid trends were communicated more widely
- The relative role of W3C, GGF and OASIS is unclear
- The difference between CORBA and the Grid and the uptake of CORBA lessons by the Grid was often commented on
- The Semantic Grid has clarified the relation of Semantic Web and Grid technologies; how do Digital Library technologies and Agents fit into the Grid?
- e-business e-government and e-military are not systematically covered

2.5 Summary of Discussion in Sections 7, 8 and Appendix

Here we summarize in one place, the Grid services and features discussed in the following detailed discussion. Section 7 is the survey of Grid Services with world-wide scope; section 8 is the summary of gaps from the UK survey; the appendix is section 14 with further detail on UK Grid services. The table lists the section numbers where information can be found.

2.5.1 Topics 1-6: Grid Types to Meta-data Services

Topics	Sect 7	Sect 8	Appendix
1: Types of Grid	**7.1**	**8.1.1, 8.1.2**	
R3	7.1	8.1.8	A.2.1.4.1
Lightweight	7.1	8.1.3, 8.1.5	A.2.1.1, A.2.1.3
P2P	7.1	8.1.4	A.2.1.2
Federation and Interoperability	7.1	8.1.6, 8.1.7	A.2.1.4.2
2: Core Infrastructure and Hosting Environment	**7.2**	**8.2.1, 8.2.2**	
Service Management			
Component Model	7.2	8.2.3.2, 8.2.3.4	A.2.2.1.2, A.2.2.1.4
Service wrapper/Invocation	7.2	8.2.3.1	A.2.2.1.1
Messaging	7.2, 7.5	8.2.3.3, 8.2.3.4	A.2.2.1.3, A.2.2.1.4
3:Security Services	**7.3**	**8.3.1, 8.3.2**	**A.2.3.1**
Certificate Authority	7.3		A.2.3.1
Authentication	7.3		A.2.3.1, A.3.3.5, A.2.9.1.7
Authorization	7.3	8.3.4.1	A.2.3.1, A.2.3.2.1, A.2.9.1.7
Policy			A.3.3.5
4:Workflow Services and Programming Model	**7.4**	**8.4.1, 8.4.2**	
Composition/Development	7.4	8.4.1.1	A.2.4.1.3, A.3.4.1.7, A.7.2.3, A.7.3.2.7
Languages and Programming	7.4	8.4.1.1	A.2.6.1.2, A.3.3, A.4.1.1, A.7.2.6
Compiler	7.4	8.4.1.2	A.2.4.1.3
Runtime Engines (Enactment)	7.4	8.4.1.2	A.2.4.1, A.7.2.2
Miscellaneous	7.4	8.4.3, 8.4.4	A.3.1
5: Notification Services	**7.5**	**8.5.1, 8.5.2**	
Miscellaneous	7.5	8.5.3	A.2.5.1, A.3.4.1.8
6: Metadata and Information Services	**7.6**	**8.6.1, 8.6.2, 8.6.3**	
Basic Metadata including Registry	7.6.2	8.6.4.1, 8.6.4.2, 8.6.4.3	A.2.6.1.1, A.2.6.1.2, A.2.6.1.3
		8.6.5.2, 8.6.5.4	A.2.6.2.2, A.2.6.2.4
Information Aggregation (events)	7.6.3		A.2.7.2.2
Semantically rich Services and meta-data	7.6.4	8.6.4.4, 8.6.4.5, 8.6.4.6	A.2.6.1.4, A.2.6.1.5, A.2.6.1.6
		8.6.5.1, 8.6.5.2	A.2.6.2.1, A.2.6.2.2, A.3.4.1.9, A.7.3
Provenance	7.6.5	8.6.5.3	A.2.6.2.3
Miscellaneous	7.6.1		A.3.3, A.7.2.3.2, A7.3

2.5.2 Topics 7 to 11: Information Grid Services to Network Services

Topics	Sect 7	Sect 8	Appendix
7: Information Grid Services	**7.7**	**8.7.1, 8.7.2**	
OGSA-DAI/DAIT	7.7	8.7.3, 8.7.3.1, 8.7.5.2	A.2.7.1, A.2.7.3.2, A.2.9.2.2(d), A.7.3.2.12
Integration with compute resources	7.7	8.7.1	A.2.7.3.3, A.2.7.3.4, A3.1
Miscellaneous	7.7	8.7.4, 8.7.5.1, 8.7.5.3, 8.7.5.4	A.2.7.2, A.2.7.3.1, A.2.7.3.3, A.2.7.3.4,A.2.9.2.2(e) A.3.1, A.3.2, A.3.4.1.6
8: Compute/File Grid Services	**7.8**	**8.8.1, 8.8.2**	
Job Planning Scheduling Management	7.8.1, 7.8.2	8.8.3.2	A.2.8.1.2, A.3.4.1.3, A.3.4.1.4
Job Submission	7.8.4		A.3.4.1.1, A.3.4.1.2
Access to Remote Files, Storage and Computers	7.8.3, 7.8.5, 7.8.6	8.8.3.3	A.2.8.1.3
Replica (cache) Management	7.8.7	8.8.3.1	A.2.8.1.1
Virtual Data	7.8.8		
Parallel Computing	7.8.9		
Job Status	7.8.10		
9: Other Services including	**7.9**	**8.9.1, 8.9.2**	
Grid Shell	7.9.1		
Accounting and Grid Economies	7.9.2	8.9.3.1	A.2.9.1.1, A.2.9.1.6
Fabric Management	7.9.3	8.9.4.1	A.2.9.2.1, A.2.9.2.2(a)
Visualization and Data-mining	7.9.4	8.9.4.4, 8.9.4.5, 8.9.4.6	A.2.9.2.2(c), A.2.9.2.5, A.2.9.2.6, A.2.9.2.7
Computational Steering	7.9.4	8.9.4.5	A.2.9.1.3, A.2.9.2.6
Collaboration	7.9.5	8.9.3.2	A.2.9.1.2
Packaging	7.9.6		
Other Technologies	7.9.7	8.9.4.2, 8.9.4.3	A.2.9.1.5, A.2.9.1.7, A.2.9.2.2(b), A.2.9.2.3, A.2.9.2.4
10: Portals and Problem Solving Environments	**7.1**	**8.10.1, 8.10.2**	
Miscellaneous			A.2.6.1.6, A.2.9.1.4, A7.3
11: Network Services	**7.11**	**8.11.1, 8.11.2, 8.11.3**	
Performance	7.11.1	8.11.3.1	A.2.11.1
Operations	7.11.2	8.11.3.2	A.2.11.1
Reservation	7.11.3	8.11.3.3	A.2.11.1

Part II: e-Science and its Technologies

3 e-Science

e-Science refers to science that is enabled by the routine use of distributed computing resources by end-user scientists [UKeS-A] [UKeS-B] [UKeS-C]. Although e-Science can be used as a methodology by individual scientists, it is most effective when used to enable distributed global collaborations involving large numbers of people and large-scale resources. e-Science makes such collaborations more productive by breaking down barriers to communication and interaction, and by making valuable resources more accessible, both to people and other computing systems. Thus, the end result is better science.

e-Science is one of a number of broad application areas that are supported by the service-oriented Grid infrastructure [Berman03B] described in Section 5. Other examples include e-Business, e-Health, and e-Government. Developments in the field of meta-computing over the past decade have had a significant impact on the emergence of e-Science. e-Science can also be seen as having developed from the field of computational science. Since the late 1980's computational science has become established as a third avenue of scientific discovery, alongside the theoretical and experimental methodologies, based mainly of large-scale computer simulations in areas such as physics and chemistry. e-Science subsumes this role, but also goes further by focusing not only on compute-intensive simulations, but also on the remote use of large-scale data and knowledge repositories, scientific instruments and experiments, and sensor arrays. Furthermore, in e-Science these distributed resources are typically used collectively to enable a new collaborative style of scientific endeavour that is changing the research ethos and driving research agendas in many areas. This broadening of scope from computational science to e-Science has led to the inclusion of fields such as bio-informatics, environmental science, and medical applications, in addition to the physical sciences.

To illustrate the range of applicability of e-Science some typical modes of use will now be considered by looking at some ongoing projects. Further information about Grid and e-Science projects can be obtained through the project URLs given in Section 15.

3.1 Particle Physics EDG and GridPP

The European DataGrid (EDG) project [EDG-C] involves researchers from several European countries. Its main aim is to design, implement, and exploit a large-scale data and computational Grid to allow distributed processing of the huge amounts of data arising in three scientific disciplines: high energy physics, biology, and Earth observation. These disciplines all have a common need for distributed, large-scale, data-intensive computing. The EDG project has an application bias focusing on the rapid development of testbeds, trans-national data distribution, and the demonstration of applications under production operation. Our analysis only studied the particle physics side of the EDG where it overlaps with goals of the UK GridPP [GridPP]. The GriPhyN project in the United States tackles a similar problem area, but over a longer time and with more emphasis on computer science research. Other US Projects iVDGL [iVDGL] and PPDG [PPDG] are addressing the nearer term data processing problem. Collectively with GriPhyn, they form the Trillium project [Trillium].

The Large Hadron Collider (LHC) [LHC] at CERN will become operational in 2005. The computational and data processing requirements of LHC experiments will be the main focus of the high energy physics component of the EDG project. The LHC will generate many petabytes of data that will require very large computational capacity to analyse. The LHC experiments will typically involve hundreds or thousands of individuals in Europe, North America, and Japan. The data volumes are so large that the data cannot be replicated at all the sites involved, nor can the data be distributed statically. Thus, collaborative access to dynamically distributed data is a key aspect of the EDG project [EDG-C]. The long-term aim is to do the LHC data processing in a number of large regional centres and the EDG will serve as a prototype

implementation of this distributed computing environment. The latter forms the LHC Computing Grid [LCG].

3.2 AstroGrid

One of the central concepts of the AstroGrid [AstroGrid] is the idea of a "virtual observatory" that allows astronomers to remotely access astronomical observatories and the enormous volumes of data that they generate. The European Astrophysical Virtual Observatory [EAVO] and the US National Virtual Observatory [NVO] are related virtual observatory projects with an international alliance [iVOA].

The AstroGrid project is mainly concerned with the management of and access to large volumes of astronomical data. Access to remote numerical computing power for large-scale simulation is not a focus of the project. Astronomical facilities that will come online in the next few years will lead to an explosion in data volume. Examples include the Wide Field Infrared Camera (WFCAM) that will be the most capable infrared imaging survey instrument in the world when it is commissioned in 2003, and the Visible and Infrared Telescope for Astronomy (VISTA) that will be in use by 2006. These types of instrument are capable of generating hundreds of gigabytes of data every night that will soon result in petabyte-scale databases. AstroGrid is motivated by the need to develop tools, techniques, and infrastructure to address the data handling problems that arise in the use of these very large astronomical databases. To this end the AstroGrid project will develop a data grid linking key astronomical databases and end users, and a suite of data mining and analysis tools for accessing, exploring, and interpreting the data. An important goal of the project is to make the federate the databases, in the sense that it will be possible to access them simultaneously and seamlessly, ideally through a single, easy-to-use interface. More details can be found in Section 8.7.4.2.

3.3 NERC DataGrid

This [NERCDataGrid] environmental science Grid falls in the class of what we call Information Grids in Section 5 and is discussed in detail in Sub-section 8.7.4.1 with other environmental and earth science Grids. The data structure of GODIVA (an oceanography Grid linking Information and Compute/File Grids) [Godiva] is described in the appendix A.2.9.1.4.

3.4 DiscoveryNet

The DiscoveryNet project [DiscoveryNet-A] has a rather different aim than those of the other projects, in that it is focused on high throughput. It aims to design, develop and implement an advanced infrastructure to support real-time processing, interpretation, integration, visualization and mining of massive amounts of time critical data generated by high throughput devices. The project covers new technology devices and technology including biochips in biology, high throughput screening technology in biochemistry and combinatorial chemistry, high throughput sensors in energy and environmental science, remote sensing and geology. A number of application studies are included in the pilot – analysis of Protein Folding Chips and SNP Chips using LFII technology, protein-based fluorescent micro array data, air sensing data, renewable energy data, and geohazard prediction data.

Discovery Net provides a service-oriented computing model for knowledge discovery, allowing users to connect to and use data analysis software as well as data sources that are made available online by third parties. In particular, Discovery Net defines the standards, architecture and tools that:

- Allow scientists to plan, manage, share and execute complex knowledge discovery and data analysis procedures available as remote services.
- Allow service providers to publish and make available data mining and data analysis software components as services to be used in knowledge discovery procedures.
- Allow data owners to provide interfaces and access to scientific databases, data stores, sensors and experimental results as services so that they can be integrated in knowledge discovery processes.

The appendix in section A.7.2 has substantially more detail on DiscoveryNet.

3.5 myGrid

The goal of myGrid [myGrid-A] is to design, develop and demonstrate higher level functionalities over an existing Grid infrastructure that support scientists in making use of complex distributed resources. An e-Scientist's workbench will be developed in this project. The workbench, not unlike that of Comb-E-Chem, aims to support: the scientific process of experimental investigation, evidence accumulation and result assimilation; the scientist's use of the community's information; and scientific collaboration, allowing dynamic groupings to tackle emergent research problems.

A novel feature of the proposed workbench is provision for personalisation facilities relating to resource selection, data management and process enactment. The myGrid design and development activity is driven by applications in bioinformatics. myGrid is developing two application environments, one that supports the analysis of functional genomic data, and another that supports the annotation of a pattern database. Both of these tasks require explicit representation and enactment of scientific processes, and have challenging performance requirements.

The appendix in section A.7.3 has substantially more detail on myGrid.

3.6 Comb-e-Chem

The Comb-e-Chem project [CombeChem] is concerned with the synthesis of new compounds by combinatorial methods. It is a collaboration between the Universities of Southampton and Bristol. The university consortium is working together with Roche Discovery, Welwyn, Pfizer, and IBM. Combinatorial methods provide new opportunities for the generation of large amounts of original chemical knowledge. To this end an extensive range of primary data needs to be accumulated, integrated and relationships modelled for maximum effectiveness. The project intends to develop an integrated platform that combines existing structure and property data sources within a grid-based information-and knowledge-sharing environment. The first requirement for this platform is to support new data collection, including process as well as product data, based on integration with electronic lab and e-logbook facilities. The next step is to integrate data generation on demand via grid-based quantum and simulation modelling to augment the experimental data. For the environment to be usable by the community at large, it will be necessary to develop interfaces that provide a unified view of resources, with transparent access to data retrieval, online modelling, and design of experiments to populate new regions of scientific interest. The service-based grid computing infrastructure required will extend to devices in the laboratory as well as data bases and computational resources.

The appendix in section A.3.2 has detail on Comb-e-Chem services.

3.7 RealityGrid

The goal of this pilot project [RealityGrid] is to enable the realistic modelling of complex condensed matter systems at the molecular and mesoscale levels, and to provide the setting for the discovery of new materials. Integration of high performance computing and visualisation facilities are critical to this pilot, providing a synthetic environment for modelling the problem that will be compared and integrated with experimental data. The RealityGrid involves the active collaboration of industry: AVS, SGI and Fujitsu are collaborating on the underlying computational and visualisation issues, Schlumberger and the Edward Jenner Institute for Vaccine Research will provide end-user scientific applications to evaluate and test the environment and tools produced by the project.

4 e-Business, e-Government, and e-Services

e-Business is concerned with the streamlining of distributed business processes. This is distinct from e-Commerce, which is the buying and selling of goods via the Internet, and from the use of e-Science by the R&D sections of industrial and business organizations. It is the strategic approach that unites all steps in the business cycle, from initial product design and the procurement of raw materials, through production, shipping, distribution, and warehousing until a finished product is delivered to a customer. However, e-Science has adopted many e-Business concepts and technologies – primarily the use of services for providing access to resources, and workflow for describing and managing distributed applications. Just as

the infrastructure for e-Science supports the formation of virtual organizations, e-Business can be used to enable "extended enterprises" that allow businesses to interact across organizational boundaries. A typical example of an e-Business application is supply chain management in which all steps in the business cycle are unified, from initial product design and the procurement of raw materials, through to production, shipping, distribution, and warehousing, until a finished product is delivered to a customer. The aim is to achieve high efficiency in process and information flow across the trading partners involved in the supply chain by managing inventory, scheduling labour, optimizing delivery, and improving productivity.

The aim of e-Government is to transform relations between government and citizens, businesses, and other arms of government through the use of Web technologies. Better delivery of government services to citizens, improved interactions with business and industry, and citizen empowerment through access to information are key aspects of e-Government. In many ways a national government is an extended enterprise, and so in this respect e-Government can be regarded as a form of e-Business, and these two areas share many concepts and technologies.

e-Science, e-Business, and e-Government are different application areas that can be based on essentially the same underlying Grid infrastructure. The different areas may emphasize different aspects of the Grid, for example, in e-Business autonomic computing that provides fault tolerance and quality of service is often more important than high performance, which features more prominently in e-Science. However, all three areas are based on the service abstraction in which every resource is represented as a service whose capabilities and interface can be dynamically discovered by potential users. Another common feature of all three areas is the widespread use of XML for describing services and for mediating their interactions.

To illustrate the use of Grid technologies outside the e-Science arena a number of different Grid scenarios will now be presented. Further information about the projects mentioned in this section can be obtained from the citations and the project descriptions in section 15.3.

4.1 Campus Grids

Grids are naturally applicable to run institutional and enterprise facilities and this is a major focus of IBM Grid efforts. Two examples were discussed for this report; the enterprise Grid for Capital Radio is discussed in section 4.3 and has a peer-to-peer structure. Here we focus on the White Rose Campus Grid [WhiteRose] which links computing resources at three universities: Leeds, Sheffield and York. This linkage of multiple heterogeneous resources promises more effective use of facilities. This Grid could also use P2P approaches for linking some resource and users but initially the major effort is a "Compute/File" Grid based on GT2. The focus is not "cycle scavenging" on "lightly used machines" (the Desktop Grid) but rather access to a few significant resources. Information Grid support could also be important. This type of installation has responsibility for a broad range of applications and so user support is a critical issue and is made particularly difficult by the rapidly changing Grid technology. Portals – seamless access to multiple resources – haves been identified as important. Commercial support for Grid technology would be attractive for this type of deployment. Security and networking are important issues for this of Grid which is pioneering the linkage of resources so that the combined facility adds value to constituent campuses.

4.2 Military and defence grids

Information Superiority and Decision Dominance are at the heart of new military thinking about the conduct of modern warfare. For example, Network-Centric Warfare [Netwarfare] "derives its power from the effective linking or networking of the warfighting enterprise." Joint Vision 2020 [JV2020] emphasises the importance of collecting, processing and disseminating an uninterrupted flow of information while exploiting or denying an adversary's ability to do the same. The UK has recently announced its Network Enabled Capability initiative with the aim of enhancing military capability through the better exploitation of information across the battlespace. As recent world events have shown, multi-national coalitions are playing an increasingly important role in military operations. Indeed, military coalitions are archetypal dynamic virtual organisations that have a limited lifetime, are formed from heterogeneous 'come as you are' elements at short notice, and need secure and partial sharing of information.

A common requirement across these programmes is the need to inter-operate and integrate heterogeneous distributed systems and to work with large volumes of information and high data rates. In these respects, they could benefit substantially from Grid computing concepts. However, security, resilience, flexibility and cost effectiveness are key considerations for the deployment of military Grids. It is also likely that there will be the need for multiple Grids supporting different aspects of the military enterprise, e.g. 'heavyweight' Grids for imagery data and 'lightweight' ubiquitous Grids running on the PDAs of military commanders in a headquarters—these Grids will need to be interoperable.

Currently, there are a number of US military programmes exploring Grid technologies in the context of Network-Centric Warfare, for example Joint Battlespace Infosphere [JBIGrid], Expeditionary Sensor Grid [ExpSensorGrid] and the Fleet Battle Experiments [FleetGrid]. In addition, the Coalition Agents Experiment [CoaxGrid] demonstrated how an agent-based Grid infrastructure could support the construction of a coherent command support system for coalition operations.

The appendix in section A.5.2 has more detail on military and defense Grids.

4.3 Capital Radio Grid

Capital Radio is the UK's leading commercial radio group [CapitalRadio], with greater revenues and profits than any other commercial radio company. This is achieved through a total of 20 analogue radio licences broadcasting to over half of the UK's adult population. These 20 analogue licenses have a near one-to-one correspondence with physical radio studios located across the length and breadth of the UK. Although Capital Radio Group has a large number of separate offices within the country, each radio station has operations that are both centrally and locally directed. There are specific sites that perform specialised functions for the group where their actions can affect the behaviour of sub-group of offices. For example, the London office is responsible for sales of National advertising campaigns. These sales affect the commercial airtime of local radio stations. Another example would be of a single radio station promoting a new artist or new material that is picked up by other stations. These are two simple but common examples that illustrate that the flow of information around the group is multidirectional. A central server solution is inappropriate due to quality of service requirements. First, the stations need to be able to operate in isolation even in the event of network or server failures. Secondly, many data transfers are time critical that could not be met by a centralised architecture in peak conditions. These requirements to have collaborating islands of networks have lead to the consideration of data grid technologies to solve these problems.

4.4 Bioinformatics Grids

Bioinformatics Grids clearly are very important in both academic government and commercial communities. The myGrid [myGrid-A] and DiscoveryNet [DiscoveryNet-A] projects, the EBI information resource at Hinxton [EBI] and European efforts coordinated at CERN [Jones03A] were surveyed by this report for Bioinformatics. This is a field which has exploited the web for many years with the Biology workbench (originally NCSA and now SDSC [BiologyWB]) being an early exemplar of web-access to a rich suite of database and simulation tools. These five year old capabilities build on communal databases set up some 20 years ago. The field can straightforwardly migrate to Web services with an obvious transition to Grid services. For example EBI considers that candidate technologies exist for almost everything they want to do with a goal of being able to traverse Life Sciences information space easily. As an initial step EBI has built SOAP interfaces to databases and portal interface at client end. Interoperability is a key problem as heterogeneity is universal. There are many autonomous Bioinformatics sources around the world with for example some using Excel; and some Oracle.

The field has developed very sophisticated "filters" (often in slightly archaic technologies like Perl) which allow much richer views of the databases than that provided by basic database query languages. These filters need to be converted to Web Services and integrated by workflow engines with the databases (converted to OGSA-DAIS [OGSA-DAIS]). myGrid is tackling the issue of using ontologies to annotate the many different Bioinformatics resources. One needs metadata to specify the provenance and reliability of data. There are important issues with evolving schemas and legacy data/schemas with could change many times per day. Further one needs to support the clean-up of the often rough raw data. Security needs

attention with companies particularly concerned about need to keep their access patterns private. Further there must be a clear versioning model so claims of prior discovery can be reliably addressed.

4.5 Distributed Aircraft Maintenance Environment (DAME)

The theme of DAME project [DAME] is the design and implementation of a fault diagnosis and prognosis system based on the Grid computing paradigm and the deployment of Grid services. In particular DAME focuses on developing an improved computer-based fault diagnosis and prognostic capability and integrating that capability with a predictive maintenance system in the context of aero-engine maintenance. The term "predictive maintenance" implies that there is sufficient time interval between the detection of a behaviour that departs from normal and the actual occurrence of a failure. The DAME system will deploy Grid services within this time window to develop a diagnosis of why an engine has deviated from normal behaviour, to provide prognosis (understanding what will happen) and to plan remedial actions that may be taken a safe and convenient point when the impact of maintenance is minimized. A central challenge of the project is to develop a proof of concept demonstrator that will address the commercial and technical requirements of the industrial partners within the consortium (Rolls-Royce and Data Systems & Solutions). The commercial considerations of the project create a set of demanding technical requirements which will be addressed through Grid computing techniques. These include:

- Transmission of high volumes of data from remote data repositories, to and from the maintenance points;
- Access to highly specialised data analysis and diagnosis software services;
- Necessity to search extremely large volumes of data.
- Virtual Organisations (VO) where the various members are located in various parts of the world and where complex interactions among multiple agents or stakeholders are needed and are facilitated by connection through the Grid
- The need to provide supporting or qualifying evidence for the diagnosis or prognosis offered;
- Addressing the business critical aspects of the commercial deployment, developing a Grid enabled system that can meet stringent dependability requirements, including Quality of Service and Security issues.

The appendix in sections A.3.1, A.5.1.1 and A.7.1 has substantially more detail on DAME.

4.6 Grid Enabled Optimisation and Design Search for Engineering (Geodise)

Engineering design search and optimisation is the process whereby engineering modelling and analysis are exploited to yield improved designs. Intelligent search tools will become a vital component of all engineering design systems and will steer the user through the process of setting up, executing and post-processing design search and optimisation activities. Such systems typically require large-scale distributed simulations to be coupled with tools to describe and modify designs using information from a knowledge base. These tools are usually physically distributed and under the control of multiple elements in the supply chain. Whilst evaluation of a single design may require the analysis of gigabytes of data, to improve the process of design can require assimilation of terabytes of distributed data. Achieving the latter goal will lead to the development of intelligent search tools.

The focus of the Geodise project [GEODISE] is on the use of computational fluid dynamics with BAE Systems, Rolls Royce, and Fluent (world leading developers of CFD codes). Geodise is being developed by the Universities of Southampton, Oxford and Manchester in collaboration with other industrial partners working in the domains of hardware (Intel), software (Microsoft), systems integration (Compusys), knowledge technologies (Epistemics), and grid-middleware (Condor) [Condor].

In summary, design optimisation needs *integrated services* shown in fig. 4.1.

- Design improvements driven by CAD tools coupled to advanced analysis codes (CFD, FEA, CEM etc.)

- On demand heterogeneous distributed computing and data spread across companies and time zones.
- Optimization "for the masses" alongside manual search as part of a problem solving environment.
- Knowledge based tools for advice and control of process as well as product.

Geodise will provide grid-based seamless access to an intelligent knowledge repository, a state-of-the-art collection of optimisation and search tools, industrial strength analysis codes, and distributed computing and data resources.

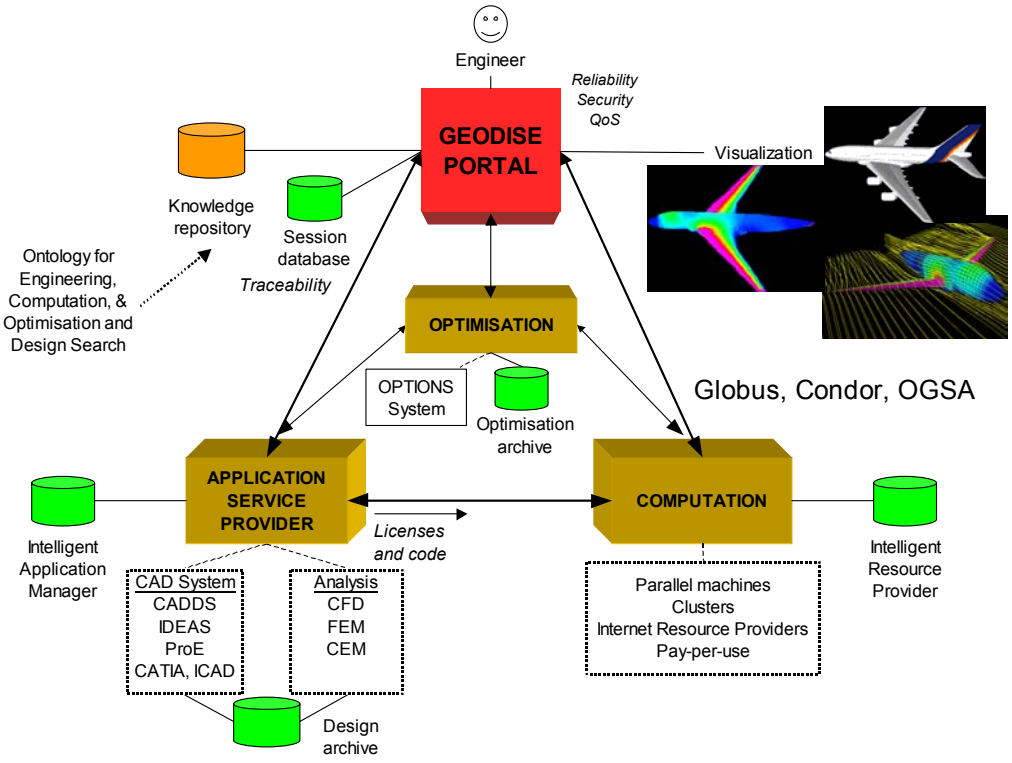


Figure 4.1: Geodise system architecture.

The appendix in sections A.3.4 and A.5.1.2 has substantially more detail on Geodise.

5 Grid and Cyberinfrastructure Technologies

5.1 Basic architecture Principles

We will use the term Grid technology [Foster99A] [Berman03A] to describe the e-Science [UKeS-A] electronic infrastructure. In the USA some would agree with this terminology but others could use "Cyberinfrastructure" [NSF03A] to describe this. We need the technology to support distributed virtual organizations [Foster01A] with a mix of shared data and compute resources accessed both in asynchronous and synchronous mode. One must support high performance data and compute capability with matching communication networks and protocols. This requirement must be combined with high functionality interfaces for users and between Grid components. The emerging Grid infrastructure addresses these stringent and often conflicting goals by using a multi-tier architecture with a range of technologies.
The current e-Science (Grid) infrastructure integrates (inherits) ideas and capabilities from many areas. These include: the Web [W3C]; Peer-to-peer (P2P) Networks [Oram01A] [JXTA]; distributed objects such

as CORBA, Java/Jini [Jini], and COM; High Performance Computing activities such as Globus [Globus-A], Legion [Grimshaw03A], Condor [Condor], NetSolve [Agrawal03A], and Ninf [Ninf] [Nakada03A]. One can consider Web services as integrating Web and distributed objects and Grid Services as linking Web services with high performance computing technologies. The architecture is built in terms of users, services and resources which are illustrated in Fig. 5.1 for both P2P networks and a more traditional multi-tier architecture.

5.2 Meta-data Rich Web Services Communicating by Messages

The UK e-Science program "projects" reflect the multiple heritages of Cyberinfrastructure and currently reflect either the "Globus" or "Web Service" (distributed object) heritage. Of course the purpose of the Gap Analysis is partly to help the integration of these heritages into a common robust Grid Service environment. As we do this, it is necessary to carefully support capabilities at each tier in fig. 5.1(a) and the communication between tiers. In discussing the needed architecture, we need to consider not only specific services but also various implicit and explicit features of the overall environment. We find it convenient to initially consider the *next generation of Grids as being based on meta-data rich Web services communicating by messages*. We believe all e-Science applications would requires this minimum capability and further one can also discuss the P2P Networks of interest to e-Science in this fashion. There is a common service based architecture with important differences in implementation of services like meta-data registration and lookup for multi-tier and P2P Grids. These services need basic support from some distributed runtime environment such as .NET [OGSI.netUVA], Jini (pure Java), Apache Tomcat/Axis (Web Service toolkit) [Axis], Enterprise JavaBeans [EJB], WebSphere (IBM) [WebSphere] or GT3 (Globus Toolkit 3) [Globus-C]. These provide the distributed equivalent of core operating system support for processes on a conventional computer.

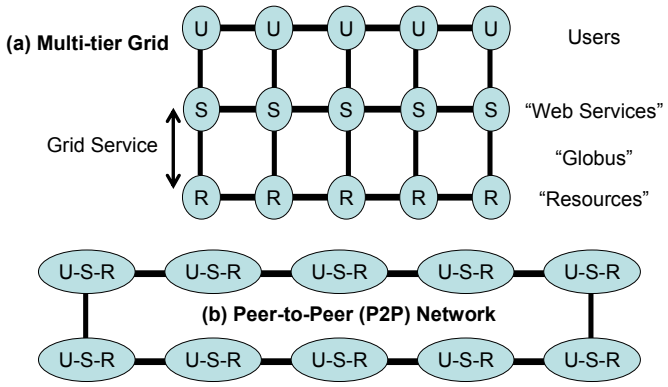
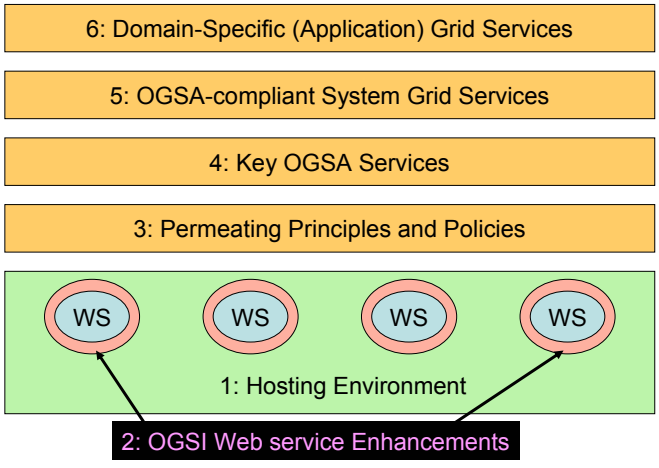


Fig. 5.1: (a) Schematic of "classic' (multi-tier) Grid showing Globus and Web Service style technologies merging to Grid Services (b) Architecture of P2P Network. U represents user; S Service; R Resource

5.3 Level 1: Hosting Environments

Fig. 5.2: Six OGSA Grid Architecture Layers



There is no precise definition of Grid architecture and this is so for many reasons. It is not easy to describe the different approaches in a single architecture framework; the key work of the OGSA working group [OGSA] at the Global Grid Forum is still in process; finally much experience needs to be gathered and processed by W3C [W3C], GGF [GGF-A], Industry, academia and government. We show in fig. 5.2 an approximate representation of how Grid systems can be divided into 6 categories in an approximate layered stack. We start with some hosting environment which can be different in different parts of a Grid and in this report defines the run-time environment for the Grid services. There

is currently no agreement on exactly what should be put in a hosting environment and other terms used are execution environment (possibly larger than hosting environment) and resource environment as discussed by the Grid Protocol Architecture (GPA) working group of the GGF [GGFGPACoreGrid].

5.4 OGSA and Level 2: OGSI Open Grid Service Infrastructure

In general there is no clear dividing line between the different layers and features can be moved between (typically adjacent) layers. The runtime layer 1 of fig. 5.2, hosts a set of enhanced Web services with the level 2 OGSI (Open Grid Services Infrastructure) enhancements [OGSI]. OGSI changes the form of Web services to support state-full components and defines special ports – notably those for notification and information. One expects the work of the W3C and GGF on web services to eventually merge with a common standard – one example is the emerging IBM-Microsoft WS-Addressing specification [WSAddressing], which allows one to "hide" differences between state-less and state-full service models.

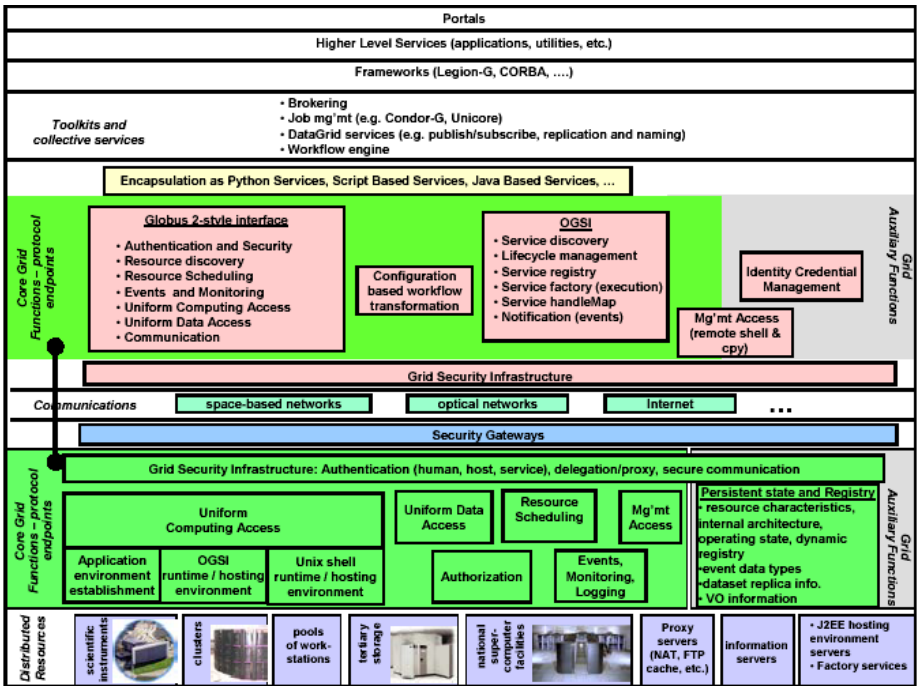


Fig. 5.3: An OGSA Grid Architecture in detail (from GGF GPA)

OGSA is designed to achieve two (related) goals. Firstly it defines common interfaces for all important services in a Grid; secondly it defines a set of common services that all Grids should offer. Levels 1 through 4 define this common set – we have used the word key in the label of level 4 to afford confusion with the OGSA core services still being discussed in the GPA [GGFGPA]. Note that a particular hosting environment might offer a certain Grid service (such as security or messaging) differently from other environments (which might not offer it at all). This is expected to be "corrected" at levels 3 and 4 so that one achieves a common set of capabilities at level 4. The complexity of a "real Grid" is illustrated by Fig. 5.3 taken from the work of the GPA.

5.5 Level 3: Permeating Principle and Policies

Level 3 of fig. 5.2 is particularly interesting as it gathered substantial comment in the Gap analysis if interpreted as enhancements to the hosting environment in the form of a set of "permeating" principles and capabilities of a Grid. These are typically not "just" services but rather features of the hosting implementation or the methods of interacting with it. We can identify ten such permeating principles of which the first is our basic Web service paradigm. The ten permeating features of a Grid are:

1) *Meta-data rich Message-linked Web Services* as the permeating paradigm

2) There could be a "User" *Component Model* such as "Enterprise JavaBean (EJB)" or .NET. If as expected we build an OGSA compliant Grid, this component model would implement OGSI features e.g. a Java EJB could have input and output channels whose XML specified interfaces were OGSI compliant.

3) The Grid requires a *Service Management* framework including a possible *Factory* mechanism to create instances from a generic service.

4) There could be a high level *Invocation Framework* describing how you interact with system components. This could for example be used to allow the system to be built from either W3C or GGF style (OGSI) Web Services and to protect the user from changes in their specifications.

5) *Security* is a service but the need for fine grain selective authorization encourages implementations where security mechanisms are present in the core runtime. For example VPN (Virtual Private Networks) are implemented at the Grid transport level.

6) A given Grid environment would be implemented differently in different situations. This can be described by a *policy context* that sets the rules for each particular Grid. Currently OGSA supports policies for routing, security and resource use.

The last four areas correspond to the lowest layers of the resource and communication capabilities.

7) The *Grid Fabric* or set of resources needs mechanisms to manage them. This includes automatic recording of meta-data and configuration of software.

8) The Grid requires reliable quality of service (QoS) for the *Network* and this implies *performance monitoring* and bandwidth reservation services. This is particularly challenging as end-to-end and not just backbone QoS is needed.

9) The Grid is built on messaging and this is built on *transport mechanisms* which can be used to support mechanisms to implement QoS and to virtualize ports and the differences between URI (Identifier) and URL (location) or between the permanent Grid Service Handle GSH and the temporary Grid Service Reference GSR.

10) *Messaging* systems like MQSeries [MQSeries] from IBM provide robustness from asynchronous delivery and can abstract destination and allow customization of content such as converting between different interface specifications.

There are other broad features such as support for provenance and notification services but it appears that these can be implemented at a higher level (4 and above in fig. 5.2) without incurring performance or functionality impairments. Here we distinguish between application level messaging (as needed for notification events) and the low level support of message transmission between components. A given Grid infrastructure may or may not make this distinction. An Autonomic Grid would need powerful low-level messaging and transport for robustness while JXTA, a well-known peer-to-peer system [JXTA], uses dynamic choices of routing and protocols to enhance peering between systems. This illustrates that some but not all differences at lower levels can be hidden by building capability at a higher level.

Virtualization is an important capability built into Grids like other advanced operating systems. In general this enables indirect specifications which focus on functionality and not implementation details. Broad-based virtualization capabilities of Grids include:

- *Location:* The URI (Universal Resource Identifier) virtualizes URL and correspondingly the GSH virtualizes a GSR
- *Protocol:* Message transport and WSDL bindings [WSDL] can virtualize transport protocol as a request that specifies the desired QoS, Security and other transport characteristics. This can in particular help transiting firewalls. Here we see the importance of virtualization of protocol and port number as port 80 has special meaning to firewalls and web services and protocols like UDP and TCP/IP are treated differently by firewalls. Originally ports and protocols were designed to specify a style of interaction between services but this is now confused as they have been "overloaded". The natural approach is to specify the desired style of interaction (one asks for low latency allowing data loss rather than the UDP protocol) and let the hosting environment (levels 1 and 3) translate this into the most appropriate protocol.
- *Publish-subscribe* messaging virtualizes the matching of source and destination addresses and is especially natural in P2P systems [Fox03C]
- The many different *brokering services* for resources virtualize resource allocation

- *Replica management* virtualizes file location and more generally one can *virtualize data* [Moore03A] as illustrated by the virtual data concept of GriPhyn [GriPhyn].
- It is at a higher level in the Grid but one way to understand the importance of the *Semantic Grid* is as the approach to *virtualizing knowledge* as a meta-data query [SemanticGrid]

We are discussing the general features of Grids that affect levels 1 through 3 of fig. 5.2 and are not easily captured as specific services. In this arena, we should mention the interplay between interfaces and protocols. Often it is not clear whether a particular specification should be a protocol or an interface and one can roughly say that protocols need to be understood by the "system" (hosting environment) but interfaces are to be interpreted by specific services. This distinction is important to produce scaling architectures so we can minimize the features needed in general system services but rather build self-learning autonomic services that have rich self-defining interfaces that can be used without system intervention. However although it is good to minimize the detailed information needed by the system, rich meta-data and semantics are critical for correct and interesting operation. In fact a major gap identified by our survey was that the current Grid activities had established the basic protocols and principles of operation (use WSDL with Grid services); however the lack of rich meta-data and the corresponding lack of Semantic interoperation is in fact the main weakness of today's Grids and Web services. The new information port in OGSI could lead to general introspection methods for discovering this needed semantic interface information.

5.6 Levels 4, 5, and 6: System and Application Grid services

Returning to Fig. 5.2, let us discuss the key Grid services of level 4 that all Grids must support. We avoid the term "core services" as this has a technical meaning [GGFGPACoreGrid] in GGF (the GPA working group) and we are not too worried about which services are in level 4 and which in level 5. Further the OGSA working group is discussing several important services but the subset of services that they will identify is not yet certain. Some critical services that can be placed in level 4 are:

- Discovery (look up) and Registration of service information (meta-data) at multiple levels including GSR, WSDL and Semantic information. It is not clear how such a multi-level look-up should be structured and if the same core database technology can be used in each case.
- Workflow or the orchestration and linkage of multiple services into a single service

These two services can be termed "global" as they must cross Grid boundaries which feature produces special challenges in building federated Grids. We can include notification in level 4 as this was part of the original OGSA paper and is well developed in the current OGSA working group. Other important services are placed in level 5 and include basic brokering/scheduling as well as meta-schedulers, OGSA-DAI (databases), the portal service (defined by WSRP - Web Services for Remote Portlets [WSRP]), provenance (interpret meta-data about history of data), sensor service for satellites and environmental or seismic instruments, job submission, file and storage interfaces, and visualization. In the presentation of the gaps, we have classified these services under broad categories such as Information and Compute/File Grids. Note that a service like OGSA-DAI is essential for Information Grids and job submission for Compute/File Grids. This illustrates that the specification of "core services" that every Grid must have is bound to be a little uncertain.

Above these system levels, we have the "real Grid" – level 6 in fig. 5.2– namely the myriad of application services with which the users interact. Here we can expect a growing activity as the lower level capabilities become clearer. All e-Science electronic capabilities should be structured as such application Web Services using portlets as their user interface; the portlets are then to be assembled as problem solving environments.

5.7 Grid Technology Suites

The best known Grid technology suite is GT2 (the Globus Toolkit 2) [Globus-B] which is largely directed at Compute/File Grids. Using the CoG (Community Grid kits) [Laszewski03A] one has been able to incorporate GT2 into Web service frameworks. GT3 (Globus Toolkit 3) [Globus-C] uses the full OGSI Grid service [OGSI] and will be phased into projects as discussed in a separate report. The success of the Java CoG kit with GT2 suggests use of GT3 will be quite smooth and in fact GT3 has adapted some CoG kit code. Given uncertainties in the evolution of Grid and W3C standards, it might be best to stick to high

level CoG kit interfaces even when GT3 is adopted. Unicore [Unicore-A] is a well regarded Grid technology used especially in Europe to support portals for Compute/File Grids. The European Data Grid [EDG-C], GridLab [GridLab] and Alien Grid [AlienGrid] are other important European Grid technology projects. There are many (promised) commercial Grids with those from IBM and Avaki having significant visibility. However the IBM Grid is not yet available and its expected heavyweight enterprise character reduces its applicability to e-Science. The Avaki [Grimshaw03A] and other commercial Grids had no experience base in the users we interviewed. Grid Systems has successfully deployed "InnerGrid" (Enterprise or Campus Grids in our parlance described in next section) of the Compute/File functionality [GridSystems].

6 Functionalities and Styles of Grids

6.1 Introduction

We can classify Grids by their technology or architecture, by their style of operation or use, and perhaps most importantly, by their function. In the first two categories we can place Semantic Grids, Collaboration Grids, Peer-to-peer Grids, Autonomic (fault-tolerant) Grids and lightweight Grids (GridLite). In Grid functions we can identify Compute and File-oriented Grids, Desktop Grids, and Information Grids. We also see many styles of Hybrid Grids mixing the above functionalities including Complexity Grids and Institutional (campus and enterprise) Grids. There are, in fact, correlations between architecture, style of use and function, but to some extent these represent independent flexibilities in categorizing Grids. Further we can compose, possibly dynamically, new Grids by joining together parts of other Grids of different functions and technologies. For example we could form a special training grid to support 20 students in a week-long class involving access to databases of some Information Grid, computers in a Compute Grid with collaboration supported by peer-to-peer technology. The different Grid services used in this case would have different scope with the security authorizations limiting the resources available in the training grid perhaps most directly defining its extent. This is illustrated in the example of fig. 6.1 with resources in four grids linked to form the training grid. Note that all 4 grids could share the same meta-data information (MDS) service and in general one has Grid architectural levels (as in fig. 5.2) and core Grid services with different scopes.

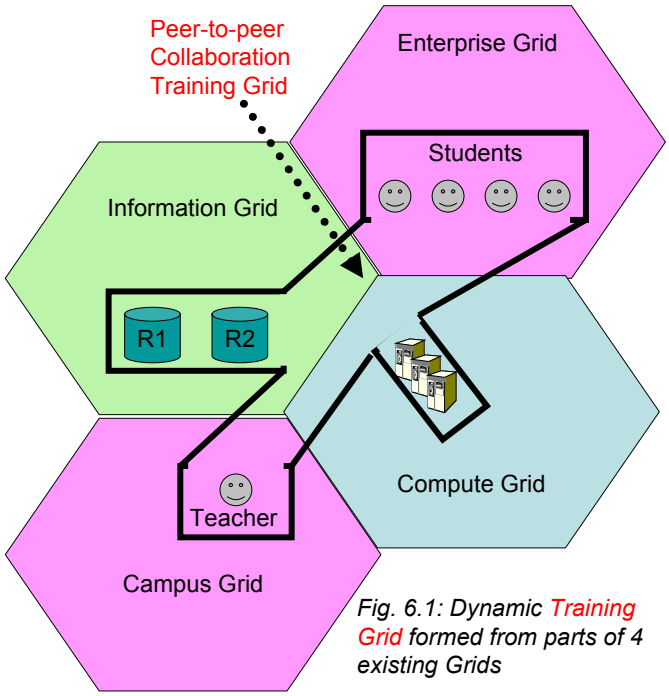


Fig. 6.1: Dynamic Training Grid formed from parts of 4 existing Grids

Note that it is not clear how grids will evolve; Will one set up multiple specialized grids each optimized for a particular task and then overlay particular virtual organizations (VOs) as above?; Will one build general purpose grids and carve out smaller VO's using security access control lists? In the action plan of part IV, we assume that one will see both models with the need to support the current multi-function model and to federate these with other "Grid Islands" of different technology and/or function.

Now we describe the major Grid functions and end this section with a detailed discussion of their overall characteristics. The types of Grids are summarized in the table below.

Type of Grid	Features
Compute/File Grid	Run multiple jobs with distributed compute and data resources
Desktop Grid	"Internet Computing" and "Cycle Scavenging" with secure sandbox on large numbers of untrusted computers
Information Grid	Grid service access to distributed repositories
Complexity Grid	Hybrid combination of Information and Compute/File Grid emphasizing integration of experimental data and simulations
Campus Grid	Grid supporting University community computing
Enterprise Grid	Grid supporting a company's enterprise infrastructure
Semantic Grid	Integration of Grid and Semantic Web meta-data and ontology technologies
Peer-to-peer Grid	Grid built with peer-to-peer mechanisms
Lightweight Grid (GridLite)	Grid designed for rapid deployment and minimum life-cycle support costs
Collaboration Grid	Grid supporting collaborative tools like the Access Grid, whiteboard and shared application.
Autonomic Grid	Fault tolerant and self-healing Grid

This table has six different functionalities followed by five styles of operation. Of course all of these can be mixed to form, say, an Autonomic Collaborative Information Grid.

6.2 Compute/File Grids

These represent the heritage of Globus GT2 [Globus-B] and Condor [Condor] to support the classic computing model of jobs running on distributed computers accessing data stored on, in general, different distributed resources. Commercially this model is of great importance to support utility computing or computing-on-demand. The initial Grid Systems InnerGrid product is in this area [GridSystems]. This model is also needed by particle physics (LHC) where the data is many petabytes per year of individual events that need to be analyzed independently and then looked at collectively to find signals of new science or to measure cross-sections. We note that the Globus team calls this a data grid and so to avoid confusion with database-centric applications highlighted in UK e-Science, we choose the compute/file grid label. We will later point out that this style of grid is, in fact, needed by applications like Bioinformatics which need to fetch sequences from a database (an Information Grid) and analyze them on dynamically allocated compute resources. The functional capabilities needed by compute/file grids are well understood and illustrated by the work of the European Data Grid (EDG) [EDG-A] and the trio of projects in the US Trillium consortium [Trillium]. These capabilities include generally important functionalities including Grid information systems (MDS-2 today), security and network monitoring discussed in section 7. Characteristic of this style of Grid are resource brokering both within collections of computers and the meta-schedulers and planners between separately managed computer subsystems. Some variant of distributed file and storage (tape) systems are needed as is the ability to create and manage data replication (caching). Fabric management – the reliable deployment of software on the tens of thousands of resources on such grids is also very relevant. Finally such grids also needed to be very robust (to cope with the inevitable glitches in data analysis on LHC scale) and so such grids need good management frameworks to deliver autonomic characteristics. Note that Compute/File Grids can in fact access their information from databases (as illustrated by the use of Objectivity in some particle physics experiments) but still the computing style is that of the classic file-based model.

6.3 Desktop Grids

Desktop Grids are the current terminology for Internet or "cycle scavenging" systems like Entropia [Chien03A], United Devices [UnitedDevices], Parabon [Parabon] and SETI@Home [SETI]. These were previously termed peer-to-peer computing systems but this term fell out of favour. Desktop Grids can be thought of as a special case of Compute/File Grids with some simplifications and some special features and needed enhancements. These Grids must cope with very heterogeneous unreliable large compute pools; further they must provide a very secure environment on each client to minimize the chance of either the downloaded application harming the client or the client creating erroneous results. In fact the commercial

systems are now targeting enterprise customers and we can expect Desktop and compute/file Grids to merge in some sense into a spectrum of "utility computing" Grids. Furthermore in this category as discussed for InnerGrid [GridSystems] above, one can expect IBM and other major enterprise vendors to offer Grid-based products addressing the dynamic assignment of the large enterprise computing software products to pools of resources used to cope with computing "hot-spots".

6.4 Information Grids

Information Grids are typified by applications like the virtual observatory and bioinformatics where the typical service is accessing a database. These grids start with a service-centric view as this approach is already popularized by the well developed web interface to databases. Correspondingly in the UK e-Science activities, projects of this kind have tended to build from a Web Service, rather than Globus, base technology. Of course with GT3 [Globus-C] and Grid services, these technologies are unified. In the fig. 6.2 below we depict the typical multi-tier view of a Grid and overlay what we term the "resource" "service" and "client" views. One of the challenges in building the next generation of grids is putting the needed capabilities in the appropriate view with the "service view" having richer functionality but typically lower bandwidth than the "resource view". We will build Grids with both "horizontal" (within each view) and vertical (intra-view) connectivities and supporting this complexity is a major challenge of Grid Programming Environments. In this terminology, current information Grids focus on the service view while Compute/File Grids are largely resource view driven. As the field matures this distinction will disappear and one will add the rich middleware needed to increase the robustness and functionality of Compute/File Grids; correspondingly Information Grids will add high performance back-end (resource view) computing and data transfer to the service view.

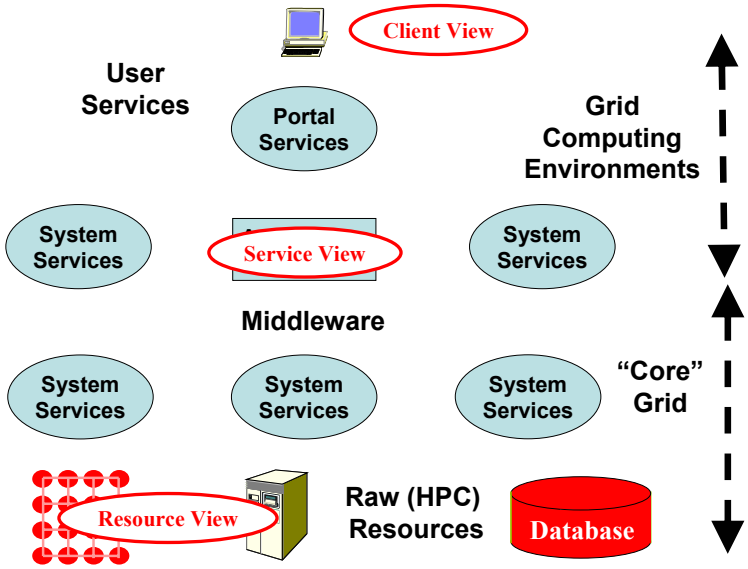


Fig. 6.2: Multi-tier Grid Architecture showing Client, Service and Resource Views

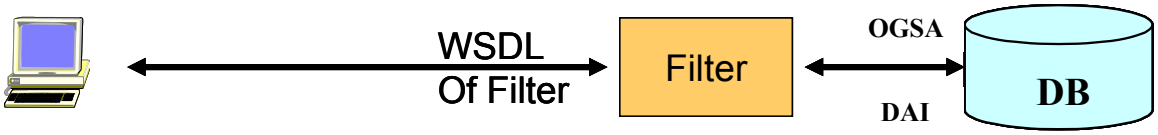


Fig. 6.3 Integration of Computing and Database Resources

OGSA-DAI is a highlight of the UK e-Science program and is the key Grid enabling technology in this area [OGSA-DAI]. As well as this service, Information Grids require basic registry and information services with rich meta-data required to annotate both the data and the different resources. The latter requirement suggests that Semantic Grid [SemanticGrid] could be useful as myGrid's approach to Bioinformatics [myGrid-A] is investigating. Compute/File Grids use workflow but typically at the resource view whereas it is often needed at the Service level in Information Grids. This is illustrated by DiscoveryNet [DiscoveryNet-B] and myGrid [myGrid-B] and is being investigated by OGSA-DAI with BPEL4WS and the earlier WSFL being well-known industry efforts [BPEL4WS] [WSFL].

As we discuss below, one often wants to filter data from repositories as in figure 6.3. This could "just" be a fixed filter specified by its WSDL interface and so retaining the pure middle-tier service model. Alternatively this filter would invoke the resources of a compute/file grid to perform the transformation; this scenario is described below as a Complexity Grid.

Information Grids include the cases where the information is gotten from Sensors (as in Earthquake Grids from SCEC [SCECGrid] or SERVOGrid [SERVOGrid]) or experimental devices (as for instance in NEESGrid [NeesGrid-A]). These real-time streams would also usually be filtered as in figure 6.3 above.

6.5 Complexity Grids

We have already noted that many applications need to combine the features of Information and Compute/File Grids. One important e-Science combination can be termed "complexity" grids as they support the emerging fields of Geocomplexity and Biocomplexity. Figure 6.4 shows one natural way to look at the problem as generalized data assimilation where the Grid manages distributed data sources which could be databases (biology), sensor nets (environment) or streams from multiple satellites (earth science). This data can be pre-processed in a distributed fashion as it is pruned and transformed for use in large scale simulations. Note that the anticipated "data deluge" is expected to produce so much data that substantial filtering could be needed to project the data onto those components of greatest value to guide the simulation. Actually one could consider particle physics to have the general structure of fig. 6.4. The data corresponds to the raw events from the accelerator; the filters correspond to the initial processing to produce data summary tapes; the data assimilation "simulation" corresponds to the physics analysis phase.

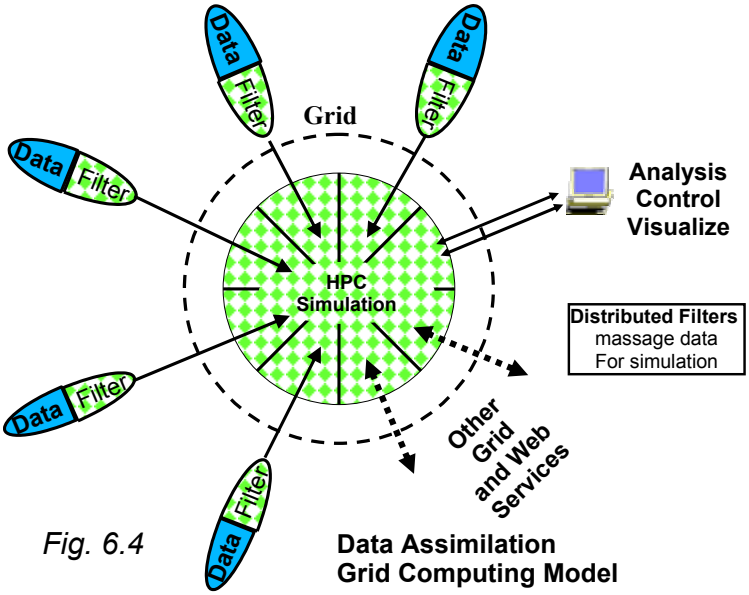


Fig. 6.4 **Data Assimilation Grid Computing Model**

In the latter case one typically needs a large parallel machine for the PDE dominated fields like climate, ocean, and weather simulations. In the particle physics case, largely uncoupled machines are necessary to support Monte Carlo simulations and data analysis.

6.6 Campus and Enterprise Grids

Complexity Grids represent one hybrid of compute and information grids emphasizing the integration of experimental data with simulations and analysis tools. The typical computing environment in any organization forms such a hybrid but can have a somewhat different emphasis in different cases. Campus Grids are nearest to Compute/File Grids while Enterprise Grids have few users running programs but need to support many large programs from vendors like Oracle and SAP. Even here there is diversity with medium-size distributed enterprises like Capital Radio (section 4.3) requiring a peer-to-peer architecture.

6.7 Grid Characteristics

Above we discussed Grids divided according to functionality but in this last subsection we divide Grids rather differently by their technology and style of operation. For the first case we consider Semantic Grids which are formed by the integration of the Semantic Web [SemanticWeb] with Grids. This involves forming ontologies and rich meta-data annotations for all resources involved in a Grid – hardware, software, data records, provenance and the repositories themselves. The Semantic web brings a process and a way (OWL the Web Ontology Language [OWL]) of expressing the meta-data. There are inference engines but perhaps these need to be specialized for the Grid which could also provide the robust OGSA-DAI federated database structure for managing the meta-data. Note that a Grid typically is interested in a relatively few high value resources whereas the Semantic web was originally aimed at very many small resources (the world's web pages); nevertheless the basic ideas of the Semantic Grid can be expected to be broadly important across all Grid functionalities [SemanticGrid].

Peer-to-peer Grids are formed from distributed systems with a service architecture that uses P2P mechanisms. These have been well described in [Oram01A] and [Fox03A]. They can involve collaboration, shared files, trust and electronic payment mechanisms. The registration and lookup mechanisms are particularly distinctive as they involve matching of messages advertising or requesting services; this is quite different from the classic database lookup and registration used in other Grid architectures. We expect P2P mechanisms to perform well in very dynamic localized situations but traditional Grids may give the best performance with very distributed less dynamic scenarios. Thus we can expect P2P Grids to be used in conjunction with other architectures and to again be broadly useful. For example in particle physics, we could perhaps use a P2P Grid in the final physics analysis which is often performed by a relatively small group exploring a set of rapidly changing hypotheses to interpret the data. The first stage of the experiment discussed as a Compute/File Grid however can use a much more structured approach.

Although many e-Science Grids will be large, some organizations will not have the resources to support the complex software stack one expects in many mainstream Grids. Thus we expect it will be valuable to develop Grids that are relatively easy to install and support throughout their life-cycle. The training example discussed in the introduction, or small university departments, are candidate users of lightweight grids. These are illustrated by JXTA [JXTA] and Jini [Jini] Java technologies which can form lightweight Grids with either peer-to-peer or conventional structure. Grid federation technology is suggested in section 8.1.7 as a way of gluing such grids together and to other more complex grids.

Collaboration is intrinsic to all grids as virtual organizations are formed around shared resources. However there are set of specialized capabilities such as audio-video conferencing, chat rooms, instant messengers and real-time shared applications. Here the Access Grid [SWOF] [AccessGrid] and systems like VRVS [VRVS] are well known academic projects while commercially WebEx [WebEx], Placeware [Placeware] and GrooveNetworks [Groove] are prominent. Clearly collaboration can be part of any grid and this style of grid operation is generally important; more details can be found in section 7.9.5.

Autonomic Grids were initially highlighted by IBM [Pattnaik03A] [Horn01A] but are now being explored broadly. Modern enterprise and distributed systems are getting larger and more complex and effective means must be found to both manage such systems and to make them fault-tolerant. A key aspect of such systems is the use of distributed control with self-monitoring and self-healing subsystems which has led to the term autonomic to describe fault-tolerant Grids of this type. There is no silver bullet to achieve autonomic behavior but there are a set of architectural principles that used together can produce this characteristic. Autonomic Grids should, for example, use asynchronous logged messaging, avoid blocking calls, impose no artificial limits (e.g., use thread pools as in Java "new I/O" NIO [JavaNIO]), employ distributed fault tolerant support systems for information, security, and registry, use scalable mechanisms, support fine-grain security, ensure that all services are restartable and have management algorithms for incomplete or erroneous processes, use component based architecture such as OGSA, have problem determination and accounting tools, and use dynamic message routing/re-routing to create an automated fail-over system.

7 Status of e-Science Service Development

In any rapidly changing field, it is hard to accurately describe the current status. Often it is hard to determine what is real and what are dreams, plans or temporary fixes; there are many projects – such as those in the UK e-Science program [UKeS-A] – which have only recently started and the documentation in papers and web pages is erratic. Thus only a process similar to that conducted in UK but on a broader scope could aim at an authoritative survey of e-Science related services world-wide. As most of the community does congregate in Grid Forum [GGF-A] meetings, it would be realistic to improve this section quite significantly by a concerted effort at a GGF meeting for example. Nevertheless we will attempt to provide a qualitative snapshot of the situation here. We will classify services in same broad areas used here for the UK programs although there are differences in emphasis – the US and European programs have more effort in Compute/File grids than the UK – which is manifested as greater level of detail in category 8 (Compute/File Grid Services) for the non UK tabulation. The broad categories are

1) Types of Grid (Autonomic, Lightweight, P2P, Federation and Interoperability)
2) Core Infrastructure and Hosting Environment (Service Management, Component Model, Invocation, Messaging)
3) Security Services
4) Workflow Services and Programming Model
5) Notification Services
6) Metadata and Information Services (Basic including Registry, Semantically rich Services and meta-data, Provenance)
7) Information Grid Services (OGSA-DAI/DAIT, Integration with compute resources, P2P and database models)
8) Compute/File Grid Services (Job Submission, Job Planning Scheduling Management, Access to Remote Files Storage and Computers, Replica (cache) Management, Virtual Data, Parallel Computing)
9) Other services including (Grid Shell, Accounting, Fabric Management, Visualization Data-mining and Computational Steering, Collaboration)
10) Portals and Problem Solving Environments
11) Network Services (Performance, Reservation, Operations)

Each category corresponds to a following sub-section and the numerical labeling is the same in sections 7, 8 and figure 2.1.

There are major projects and consortia whose work gives a reasonable – albeit incomplete – picture of the worldwide activity in Grid services. In addition to UK e-Science activities discussed in sections 3, 4 and the appendix (section 14), we have examined:

- Commercial activities especially those of IBM, Avaki [Grimshaw03A], Grid Systems [GridSystems], Platform [Platform], Sun [SGE], Entropia [Chien03A] and United Devices [UnitedDevices]
- The GT2 [Globus-B] and GT3 [Globus-C] Globus [Globus-A] Toolkits. Here we effectively covering not just the Globus team but the major projects such the NASA Information Power Grid [IPG] that have blazed the trail of "productizing" Grids [Johnston03A]. Note that we can "already" see GT3 (Grid Service) like functionality from GT2 wrapped with the various (Java, Perl, Python, CORBA) CoG kits [Laszewski03A].
- The European Data Grid (EDG) [EDG-C] which has also made major contributions to testing and enhancing Globus. Other technology oriented European projects include DataTAG [SGE], GRIP [GRIP] and GridLab [GridLab]. These are part of a cluster of 20 EU funded Grid projects that are being integrated by the GridStart project [GRIDSTART]. Alien Grid is a Grid developed for a CERN experiment [AlienGrid].
- Unicore [Unicore-A] which is the most sophisticated integrated "seamless access" job submission system.
- Trillium [Trillium] (GriPhyn [GriPhyn], iVDGL[iVDGL]and PPDG [PPDG]) focusing on development of Grid for a set of Grid applications.
- Condor [Condor] from the University of Wisconsin which is being integrated into Grid services through the Trillium activities.
- The NMI or NSF Middleware Initiative [NMI] packaging a suite of Globus, Condor and Internet2 software. This has overlaps with the VDT (Virtual Data Toolkit from GriPhyn)

- Storage Resource Broker [SRBMCAT] from SDSC
- The DoE Science Grid [Johnston03B] and related activities [Johnston03A] such as the Common Component Architecture (CCA) [CCA] project
- Examination of services from a collection of portal projects in the US from Argonne [Laszewski02A], Indiana [CGLIndiana] [ExtremeIndiana], Michigan [CHEFportal], NCSA [NCSA] [NCSAGAMS] and Texas [OGCE]. This includes best practice discussion from Global Grid Forum [Fox03B].
- Review of contributions to the recent book *Grid Computing: Making the Global Infrastructure a Reality* edited by Fran Berman, Geoffrey Fox and Tony Hey, John Wiley & Sons, Chichester, England, ISBN 0-470-85319-0, March 2003 [Berman03A].

We note Trillium and EDG have a primary focus on particle physics but have other applications which include Information Grids in astronomy [Williams03A] for GriPhyn [GriPhyn] (LIGO gravitational wave detector [LIGO] and Sloan Digital Sky Survey [SDSS]), and for EDG earth observation [EDGWP9] and biomedicine [EDGWP10]. These applications do not yet exploit the emerging OGSA-DAI framework but are largely built on compute/file model where an apparatus gathers data and creates files; these files are processed in the classic distributed UNIX Shell (Grid Shell) model underlying compute/file Grid architectures.

In the following, we discuss briefly the world wide status of Grid Services noting there is substantially more detail on the UK activities in section 8 and the appendix to our report.

7.1 Types of Grid

In section 6, we described different functionalities and styles of Grids and there are examples of all the categories we introduced although not all the relevant projects were (originally) termed Grids. Desktop Grids were developed by many commercial and academic projects with Entropia [Chien03A] and United Devices [UnitedDevices] being the most prominent commercial offerings. There are clear similarities between such systems and the powerful scheduling systems like Condor [Thain03A] discussed in sec. 7.8. "Computing-on-demand" can be implemented either with a large number of small resources as envisaged in desktop computing or with different mixes of resources including parallel computers and large servers optimized to run the major enterprise software systems. The latter are supported in current IBM Enterprise software environments and although IBM appears to view Grid ideas as important for Enterprise software, it is not clear how this translates into their software architecture; it is for instance too early for wide spread use of OGSI compatible services. The Sun Grid Engine [SGE] has also been widely deployed and supports both Campus and Enterprise Grid models with similar functionalities to Condor [Thain03A] but probably again more emphasis on a heterogeneous mix of largish machines typical of Enterprise and campus central facilities.

There is substantial experience with compute/file Grids as supported by Globus with EDG (European Data Grid) providing both technology enhancements and interesting application evaluations in their three focus areas. The particle physics area receives special attention from the EDG due to the critical need for Grid capabilities for analysis of Large Hadron Collider (LHC) data. Although EDG includes earth observation and biological applications, these are tackled using the same Compute/File methodology needed by particle physics data analysis. Note that these applications can all be called "data" grids as they support large scale experimental science. There is much debate as to best model for such problems – should one move data to the computer (where the software is) or vice versa. Many arguments suggest it is most efficient to move the computing to the data but it is not so clear that it is practical to assume that the compute resources will be next to the data as perhaps they are owned by different organizations. The UK e-Science program has emphasized Information Grids emphasizing data oriented applications but with a service not a "distributed job" paradigm. Virtual Observatory and Bioinformatics where there are also several important projects worldwide fit this model. As GT2 moves to GT3, one will tend to merge data and Information Grid models but they will be differentiated by their respective emphasis on the resource and service tiers respectively. As noted in section 6.4, we expect initial Grid systems will tend to emphasize either the resource (program) or service level depending on their heritage. We expect balanced systems with both information (service) and data levels to gradually emerge as the complexities of a multi-level architecture become both

understood and supported. Avaki has commercialized the well-known and innovative Legion [Grimshaw03A] system which pioneered distributed object technology in high performance computing. Detailed evaluations of this Compute/File Grid system were not obtained in this study.
The need for RRR (Autonomic) Grids was emphasized by the EDG experience in deploying current systems in trial runs for the LHC Computing Grid [EDGWP8HEP]. The Autonomic Grid concept has been popularized by IBM [Pattnaik03A] but there appears little consensus or experience in development of robust Grids – this contrasts with the seemingly universal agreement of the need for such Grids. This issue is discussed in the appendix, section A.2.1.4.1. Lightweight and peer-to-peer Grids are required for several applications and there are some early indication of success for this style of Grid using Jini/JXTA from Sun or .NET from Microsoft. Jini [Jini] Grids include CoABS [CoABS-A] [CoABS-B] (Darpa) and ICENI. JXTA [JXTA] has been used in the Cardiff Triana [Triana-A] [Triana-B] environment and evaluated by the Capital Radio Group for multi-media content management. The Community Grids Laboratory at Indiana University has built Grids integrating JXTA with more conventional Grid architectures [Fox03C]. The appendix has discussion of Jini (section A.2.1.1), JXTA (Section A.2.1.2) and .NET (section A.2.1.3). Federation is an attractive way of integrating Grid "Islands" built on different underlying technologies but there is currently no experience with this. One interesting concept "The Virtual Private Grid" [Pierce03A] suggests extending the successful VPN (Virtual Private Network) to dynamically build Grids.Grids and VPN's are discussed in the appendix, section A.2.1.4.2.

7.2 Core Infrastructure and Hosting Environment

The survey found substantial interest in the first three layers of the architecture stack – Hosting environment, OGSI [OGSI] component model and the permeating principles and policies of fig. 4. As discussed in section 5, it is not clearly useful to distinguish different parts of these lower levels which together represent the essential run-time environment and imply key aspects of the Grid development paradigm. Decisions in these layers affect strongly the characteristics (Peer-to-peer, autonomic and lightweight) of the hosted Grid. Correspondingly Grids built on top of different infrastructures will tend to have different natural characteristics. The most common approach is to use the standard set of Apache Java and Web service tools (Axis WSIF WSDL4J) [Axis] [WSIF] [WSDL4J]. GT3 [Globus-C] makes this choice and we already have substantial Grid experience with this infrastructure from the many Grid Computing Environments (GCE) [GGFGCE-A], Java CoG Kit [Laszewski03A], GPDK [Novotny03A] and Gateway [Haupt03A], using it. JXTA and especially Jini based Grids have been successful and not surprisingly there is significant interest in .NET with the University of Virginia leading the GGF activity [OGSI.netUVA] in this area.

OGSI defines the stateful service runtime component structure but this is somewhat independent of the development or user environment which itself should exhibit a clear object-based paradigm. Industry typically uses EJB's (Enterprise Javabeans) [EJB] as the Java component model but there are possible difficulties in adopting this for the Grid; EJB support systems tend to be heavyweight and not suitable for the typical user; further it is not clear that EJB's naturally support scientific applications. The GCE systems discussed above were typically built with lighter weight approaches such as servlets. There are two well established user component activities; the common component architecture CCA [CCA] in the USA and ICENI (appendix, section A.2.2.1.2 and [ICENI]) from the UK e-Science program. However this type of approach is still rather experimental and there is no consensus as to the important issues. The core component model combined with an approach to workflow can be thought of as the Grid Programming paradigm and as discussed in sec. 7.4, we can expect this to be an active area of both research and development mirroring the diversity of approaches explored for "ordinary programming".

The use of Interceptors and wrappers to enhance object interfaces is a well established middleware concept and this idea could be useful at this stage to provide frameworks that hide the inevitably evolving details of Grid Service interfaces. Such a Grid interaction or invocation framework (appendix, section A.2.2.1.1) could automate the generation of service instances to allow OGSI factory [Gannon03A] and pure Web Services to coexist. There is substantial new middleware research for building adaptable lightweight systems that could be exploited for the Grid. Metaobject protocol (MOP) [OpenJavaMOP] is one approach to interceptors while middleware component models include OpenCOM [Clarke01A], THINK [Fassino02A], Knit [Reid00A], Click [Kohler99A]; K-Components [Dowling03A] [Dowling02A],

LegORB and Universal Inter-Connect (UIC) [Roman00A] and JavaPod [Bruneton00A]. It is possible that new research initiatives applying such ideas to the Grid would be important. The appendix, section A.2.2.1.4, has more details.

Fault tolerance of Grid systems will depend critically on decisions made at the core architectural level and there is currently little work in robust or autonomic Grids. There is a fault tolerant CORBA [SchmidtWS-A] standard [FTCORBA] but none yet for Java or web services. One needs a powerful service management framework monitoring the performance of each service and if necessary augmenting or replacing any service. This is helped by message oriented middleware where each physical message can be tracked and if necessary stored for later delivery if a destination resource is overloaded. WebSphere MQ [MQSeries] from IBM is a well known infrastructure of this type and NaradaBrokering [NaradaBrokering] from Indiana has some useful capabilities in this area. The new proposed reliable Web service messaging standard [WSReliableMessage] [OASISWSRM] illustrates the commercial interest in this area.

7.3 Security Services

We did not study this in depth and can recommend the thorough work of the EDG in this area. The two documents [EDGWP7Secreq] and [EDGWP7Secdes] discuss respectively the requirements and architecture for a Grid Security system. Howard Chivers (appendix, section A2.3.1.1 and [Chivers03A]) has given a good description of the difficulties in Grid security and the use of a federated architecture to address it. Marlon Pierce [Pierce03A] describes how such a security architecture could be implemented using messaging systems like NaradaBrokering. The Global Grid Forum [GGF-B] has several authoritative documents. The UK e-Science program security task force [UKeSSTF] is preparing both a program wide policy and a set of reports with a useful paper by Surridge already available [Surridge02A].

Security covers several topics including authentication, authorization, auditing, privacy, non-repudiation, confidentiality, integrity, trust, assurance, manageability, delegation, firewalls, proxies and certificates. Accounting is often included in this list but we choose to group this with economies in section 7.9. We group our comments here under four broad areas “architecture”, authentication, authorization (access control) and certificates. The architecture and corresponding implementation of Grid security is expected to change as one shifts to Web service WS-security based on messaging. This should be contrasted with current implementations which tend to be transport (e.g. use SSL) based. Message based security avoids several problems with current Grid security as it isolates security issues to the message and the two end-points communicating. One expects new technologies such as SAML (Security Assertion Markup Language from OASIS) to be used in future systems. Important projects include Globus, EDG, UNICORE and Avaki/Legion in Grid community while Microsoft Passport, the Liberty Alliance [Liberty] and Shibboleth [Shibboleth] are broader based activities. We believe that firewalls can be both implemented and “tunneled” by combining good security architecture with a messaging infrastructure like Narabrokering that virtualizes transport protocol. There are some important Security topology issues such as firewall location, DMZs, hierarchical VPNs, and other (virtual and real) networking tricks that are used to reduce sensitivity to security failures. Related to this are Grid intrusion detection techniques that have not been investigated significantly.

Authentication systems, which serve as the first and primary line of defense in most Grids, have well known implementation problems in scalability and reliability. The current Grid certificate authorities are clumsy to use. Further there are multiple authentication schemes: PKI (Public Key), Kerberos and “simple password” based systems with smartcard and other enhancements. Michigan’s K509 project [MichiganKX509] Indiana’s Gateway portal [Gateway] and Sandia [Moore01A] have addressed the use of Kerberos in Grid systems as there are many organizations that require Kerberos authentication. Michigan builds a Kerberos interface to a PKI certificate system; Indiana uses Kerberos as security for a non-Globus SOAP+SAML portal while the Sandia ASCI work avoids PKI by implementing the Globus GSI in Kerberos. The MyProxy [MyProxy] Globus system is an important technology to generate the proxy certificates required in their current solution; the difficulties in this approach – especially with the expiration of temporary certificates – appear to be addressable in the Web service security framework. There is work in the portal community from the CHEF project [CHEFportal] in linking Grid and portal security systems.

Authorization is a hard problem because of the diversity of users and resources in any Grid environment. The seamless access model – seemingly key to simple powerful virtual organizations – is not directly applicable to Grids with resources of very different security levels and linking "real" organizations with very different security policies. This requires fine grain reliable authorization with a clear accountability for the authorization given to any user. The Grid community is using CAS2 from Globus [GlobusCAS] which has been enhanced to include capabilities pioneered by VOMS from the EDG [EDGWP2VOMS]. Akenti from Lawrence Berkeley Laboratory [Akenti] and Permis [Permis] (part of NSF's NMI release) are major projects in this area with a broader audience than Grids. As emphasized in the Chivers' paper quoted above federated security architecture is required for a scalable authorization scheme.

The peer-to-peer community has addressed trust and security from different points of view; these could become relevant if peer-to-peer Grid architectures become popular.

7.4 Workflow Services and Programming Model

Workflow is a critical element of any Grid for it covers the capability of linking different services together to produce new more complex services. In spite of or perhaps because of its ubiquity there is surprising little consensus as to the "right way" to implement workflow. In fact even the name is not so obvious with alternatives being "Service Orchestration", "Service or Process Coordination", "Service Conversation", "Web or Grid Scripting", "Application Integration", "Software Bus" or perhaps best "Web or Grid Programming". The different names reflect the different sources of work in this area and we are not aware of any comprehensive review of this important field; the Advanced Programming Models Research Group of the Grid Forum has a good review [Lee01A] from the distributed computing perspective; however it is does not focus on Web or Grid service based models. Marinescu has put together interesting material including a Global Grid Forum whitepaper [Marinescu02B], a book [Marinescu02A] and proceedings of a recent meeting in this area [Marinescu01A]. The Grid Computing Environments research group of the Global Grid Forum has discussed workflow with different groups giving informal presentations (at both GGF5 and GGF6) but did not identify yet enough consensus and depth of work to produce authoritative summaries or discussion of best practice. [GGFGCE-B] [Mehrotra02A]

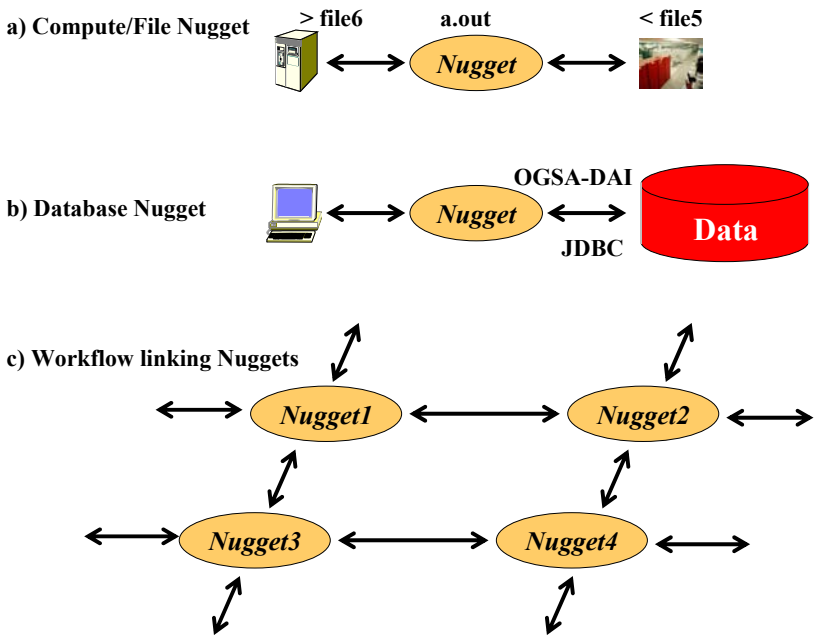


Fig. 7.1: Two-level Grid Programming Model

The term workflow comes from the important industry field of managing business processes although traditionally this has focused on applications which are more asynchronous and do not have the e-Science requirement to orchestrate large data and control flows with millisecond or lower processing times. It is in fact probably unfortunate that the name has been adopted as at least in the Grid area, the "players" and issues are rather different from those addressed by the Industry Workflow Consortium WfMC [WfMC].

There is general agreement on a two-level programming model for the Grid and more generally the Internet. At the lower level, there is "microscopic" software controlling individual CPU's and written in familiar languages like Fortran, Java and C++ . We assume that these languages generate "nuggets" or code modules and it is making these nuggets associated with a single resource that "traditional" programming addresses. To give examples illustrated in fig. 7.1, the nugget could be the SQL interface to a database, a parallel image processing algorithm or a finite element solver. This well understood (but of course still unsolved) "nugget programming" must be augmented for the Grid by the integration of the distributed nuggets together into a complete "executable" exemplified in fig. 7.1(c). Programming the nugget internals is currently viewed as outside the Grid although projects like GrADS [GrADS] are looking at integration of individual resource (nugget) and Grid programming. Typically workflow assumes that each nugget has been programmed and we "just" need to look at their integration.

Such workflow is actually quite familiar and can be thought of as generalizing "Shell/Perl…" scripts used on individual resources for UNIX operating systems and the Microsoft Com/ActiveX/…. interfaces in PC Case. Going back to fig. 7.1(a), UNIX supports very simple workflow using the pipe concept so that two compute/file nuggets a.out1 and a.out2 can be flowed together with a syntax like a.out1 <file5 | a.out2 >file6. There are a set of elementary Grid operations which generalize the traditional UNIX Shell and can be collected together as a Grid Shell described further in Sec. 7.9.1 [Nacar03A]. These Grid Shell services can be linked with other system and application services Grid workflow.

We have already noted that Workflow can be thought of as Grid Programming and there are several different possible computing models or programming paradigms even within the two-level simplification described above. One simple but powerful concept is captured by the well-known Ninf [Nakada03A] [Ninf] and NetSolve [Agrawal03A] systems which support a remote procedure call GridRPC [GridRPC] programming model. This can be contrasted with the portal based approaches [Haupt03A], [Thomas03A] where one usually does not concentrate directly on communication between the nuggets but rather separately maintains both "real" entities and separately entities representing the meta-data describing the "real" entity. We expect use of such a separated architecture to continue and indeed expand for there is a clear need to define more meta-data and it seems likely that this metadata will often be stored separately from the resource it describes. In a typical example, the Grid service containing rich meta-data might generate a batch script [Mock02A] which would cause the "real software" to execute. This approach has the advantage that no changes are required of the existing code; "wrapping a code" corresponds to capturing and putting in a separate service, the meta-data that includes all information about the codes execution and it's input/output. The Community Grids Lab has developed schema for defining application metadata and associated Application Web Services [Pierce02A]. Returning to fig. 6.2, we see the meta-data approach corresponds to the "service view" while alternative approaches such as GridRPC focus on the "resource view". We can expect future programming models to more clearly allow all levels of a Grid system to be addressed and better understanding to emerge as to "what capabilities" are best placed at "what level". Cactus [Allen03A] [Cactus] with its follow-on GridLab [GridLab] EU project has developed a sophisticated "resource view" programming model.

The Grid programming paradigm can include features of the low level Grid infrastructure and we mentioned in sec. 7.2 the Common Component Architecture [CCA] and the ICENI [ICENI] projects, which support a component-based programming model.

Workflow technology has many overlaps with the "software bus" used in many problem solving environments (PSE's) and the software used to support the linking together of multiple software modules (code-coupling or application-integration). Examples include HPC-MW [HPC-MW] PUNCH [Punch] and Pythia [Pythia] for PSE's with mpCCI [MpCCI] for code-coupling. The Purdue group of John Rice pioneered many key ideas in PSE's [Dongarra02B] and they are discussed more generally in Sec. 7.10. We expect the areas of PSE Software Integration and Grid/Web workflow to merge and be supported by a common technology suite in the future. One of the simplest and most powerful approaches to workflow and these other areas is "just" to program the module linkage with scripting (such as Python) or compiled (like Java) languages. One can of course use scientific environments like Matlab [MATLAB] and Mathematica [Mathematica] as the scripting medium and the e-Science GEODISE project has used Matlab [GEODISE]. Scripting is particularly natural for PSE's as the integration is often naturally centralized to the host

containing the script. For Grid workflow, one in general must support distributed concurrent control and communicating data between the participating Grid(web) services. This is a challenge for all workflow runtime and is particularly nontrivial for the scripting approach. Note the natural message-based interaction of Web and Grid services has historically not been generally used for PSE's but we expect this to change and message-based systems to become the standard for PSE's.

Workflow needs many capabilities and we highlight four key somewhat independent components:

- *Composition/Development:* This is the top level and corresponds to the workflow IDE (Integrated Development Environment) and often has a graph editor to be able to visually form the graph of "linked" nuggets. This editor allows one to choose from a palette of available services with ports that one can choose to link. This paradigm is familiar from visualization and image processing where systems like AVS [AVS] and Khoros [Khoros] are well established. A more modern example aimed (partly) at the Grid is the SCIRun [SCIRun]environment from Utah.
- *Languages and Programming:* There is usually an underlying formal language describing the workflow and the workflow is expressed as some sort of program in terms of it.
- *Compiler/Interpreter:* This part of the workflow system translates the result of the first two steps, composition and/or program, into the executable form.
- *Runtime Engines:* This corresponds to the components of the execution environment supporting the execution of the workflow. The term enactment is often used and this could refer to either this or the last two components (Interpreter plus runtime)

These four components correspond to related capabilities in conventional programming and so are not surprising for workflow is as we have said "just Grid programming". One might expect that different groups would focus on particular components but typically most projects appear to produce "complete workflow systems" and do not separate out the four parts highlighted above. As an exception to this, there are the activities that focus on languages and programming models. Further both Cardiff and Southampton have production standalone workflow enactment engines driven by particular languages. We expect a more modular approach will emerge as the field matures and clearer consensus develops as to the key workflow requirements and features; in fact we suggest more attention is needed on workflow runtime with attention to the possible different requirements needed for business and scientific applications. We discuss further in sections 7.2, 8.2 and 8.4 the different support needed by transaction (business) and high volume dataflow (science) based service-interaction.

As well as the direct scripting approach, several specialized workflow languages have been developed for Web services; the best known from an industry team led by IBM and Microsoft is BPEL4WS (Business Process Execution Language for Web Services) [BPEL4WS] expressed in XML. This was based on an earlier version WSFL (Web Services Flow Language) [WSFL] and has been examined in some detail by the Grid community; for example the Grid Workflow Language GSFL [Krishnan03A] was motivated by WSFL and incorporates Grid service extensions. The Cardiff Grid group has noted difficulties [Huang03] in expressing key scientific workflow concepts in BPEL4WS/WSFL and has developed an e-Science alternative SWFL (Service Workflow Language) [SWFL]. The MyGrid workflow system [DPML] uses a language similar to WSFL. Other industry XML-based workflow approaches include XPDL [XPDL] (XML Process Definition Language) from the Workflow Consortium and WSCL or Web Services Conversation Language [WSCL] where it's the nuggets having conversations and not the users! Workflow expression typically embodies traditional programming language features such as sequences of operations, flow control, and conditionals. Flow control and conditionals are not elegantly expressed in XML and perhaps a more traditional programming language syntax is more appropriate. Other examples of workflow expression languages include the GALE system [DAGMan] from Sandia National Labs, and DAGMan/Condor from the University of Wisconsin [Bivens01A]. Petri nets have been popular approach to describing systems at the "nugget level" and these are discussed by Marinescu [Marinescu02B]. There are many different workflow languages and programming models but this is not surprising from our experiences with "ordinary programming". We can expect it to be useful to have multiple workflow paradigms and multiple languages and it is unlikely that any one of these is "best". However as in "ordinary programming" one might expect different languages to share a common runtime and workflow enactment.

The workflow enactment engine implementation is one of the more complicated pieces. The engine is responsible for binding the workflow expression (using one of the languages discussed above) to the actual service components and setting up the necessary inter-service communication registrations (using registries discussed below). It must also, based on the "problem description", pick appropriate and available services (using rich metadata). The engine must then execute the workflow. Such engines may be a centralized service, or it may be implemented in a more dynamic and scalable "web of agents" fashion. We thus foresee multiple types and implementations of workflow engines. Implementation issues include fault and error handling (critical in distributed systems); concurrency control; "late binding" that selects services at runtime rather than binding up services statically; and steering, or the ability of an external observer (a human or perhaps agent) to alter the workflow during the execution. As another challenge to both the workflow language and enactment, one must take into account both needed latency/bandwidth of application and network constraints (firewalls) to decide most appropriate communication mechanism between nuggets. This typically runtime specification of the implementation of a particular service-service interaction has no agreed approach. There are of course many examples of its use with particular implementation strategies. "Agents", "brokers" and "profiles" are typical of the language one often uses to describe this adaptive mechanism. In fact it seems possible that the field of agents will merge with that of the Grid. Further this aspect of the workflow problem requires interactions between services, workflow control and the network; capabilities like the Network Weather Service NWS [NWS] covered in Sec. 7.11 are relevant here.

There have been several workflow engines developed including one from IBM to support BPEL4WS [BPWS4J]. In the job submission and scheduling area, Condor [Thain03A] and UNICORE [Romberg02A] include workflow engines of a specialized type needed for task integration in Compute/File Grids. Cardiff produced a research workflow engine JISGA [JISGA] based on Jini while an NCSA-Argonne collaboration has ingeniously extended [GridANT] the well known XML based "Java make" system ANT to support workflow. The Triana [Triana-C] project from Cardiff uses a peer-to-peer workflow architecture built on JXTA.

Any workflow runtime builds on runtime with many different capabilities. In particular, workflow must support the hierarchical composition of services to build larger megaservices and so is strongly affected by the nature of these services. We have suggested in Sec. 6, that message-based interactions and the richness of their metadata will be key hallmarks of Web and Grid services. Correspondingly these characteristic features are important for any workflow environment.

Looking at the metadata area, this is a particular emphasis of the ICENI [ICENI] component model and the bioinformatics workflow engines for the MyGrid [myGrid-B] and DiscoveryNet [DiscoveryNet-B] e-Science projects. Many Web service and Grid projects have noted that the simple metadata specification of UDDI is insufficient for reliably linking services together; this was a major conclusion [OGCWS] of the OpenGIS Consortium (OGC) when they attempted to define industry standard services for the Geographic Information Services. In the Semantic Grid, many of the semantic primitives for workflow can be derived from the ontological description of the metadata. More complicated logic rules are built on top of the ontology layer. The Semantic Grid "workflow" is expressed in another markup language (such as OWL [OWL]) and processed by software agents to find appropriate services for particular problems. DiscoveryNet has introduced another XML based workflow language DPML (Discovery Process Markup language) [DPML] to express the semantic information (ontologies) describing their services. The discussion of metadata must also take account of the common strategy described earlier of storing the metadata in the service separately from the "real" resource.

Service components in a workflow must be able to communicate to notify each other of state changes and to take appropriate actions. As mentioned one must support both control and data transfer communication with service communications that are both lightweight (small XML messages) and heavyweight (large file transfers). The workflow must interact with the messaging environment and support the natural concurrency of information streaming between distinct services. This will requires interfacing with "message-oriented middleware," which may be used for both publish/subscribe and distributed event systems like IBM's MQSeries (now WebSphere MQ) [MQSeries], Java Message Service [JMS], NaradaBrokering [NaradaBrokering] and XEvents/XMessages [Slominski02A] from Indiana University's

Extreme Lab. The runtime must deal with messaging both at the core infrastructure level (discussed in Sec. 7.2) and higher notification level (Sec. 7.5). As one moves to the OGSI standard for Grid Services, the workflow engines will need to deal with the factory mechanism and the Extreme laboratory from Indiana has described [Gannon03A] early experience with this.

As well as execution, workflow interacts with the planning and brokering phase of the Grid and with the idea of virtual data which is still at the research phase. Both of these ideas are discussed in Sec. 7.8 as although they are broadly applicable to all styles of Grids, the initial work has been developed for the Compute/File case. Executing any workflow (possibly called a job for Compute/File Grids) requires its specification (using languages described above), then the mapping of services (job tasks) to execution resources using estimates of needed capability (CPU, network, data storage). This mapping is the planning phase and requires a planning service that understands all the issues needed for the enactment. Virtual data [Chimera] [Moore03A] is the key idea of the GriPhyn project [GriPhyn] and allows one to reference data either directly (by location including caching support) or indirectly via the workflow needed to create it. This highlights the need for workflow languages to be hierarchical so that one can interchangeably use data (services) or workflow scripts in them. The DiscoveryNet workflow [DiscoveryNet-B] has this feature but it does not appear to have been generally included in workflow systems.

In summary, workflow for scientific applications is an open field in need of research and prototyping. We need to understand better current systems and develop best-practice in the different workflow components. It is notable that from the Grid to PSE's to "software/application integration", many groups are tackling roughly the same problem from different perspectives. It might be helpful if they all agreed that they were "just doing workflow".

7.5 Notification Services

Notification is an essential service in many Grid applications for this corresponds to critical high-level events (time-stamped messages) recording important state changes in services and resources. Typical Grid events include status changes for jobs (file output complete or error condition invoked) sent either to the user or more generally to a middle-tier management service controlling the job scheduling or workflow. Other events could record a change in PowerPoint presentation in an Access Grid session or that your favorite Bioinformatics database had an important update.

There are in fact a spectrum of messages at different levels of the Grid and it is not quite clear how to distinguish them and in fact if it is useful to do so. In Sec. 7.2, we noted that message-oriented middleware can manage all inter service messages and provide the basis of a fault tolerant service management framework. In section 7.6, we will discuss information aggregation web services such as the very successful HotPage [HotPage] which accumulate job status and related data but intend this as an information service and not a trigger for some action as expected for a notification events. These three message styles can be handled by the same basic infrastructure (conventionally a publish-subscribe messaging system linked to an archiving database) but differ in the way they impact services and possibly by the actual amount of data involved. More experience is needed to clarify the overlap between these different Grid events and the "best" way to handle to them.

The importance of the Grid Notification service has been highlighted by its inclusion in both OGSA [OGSA] and OGSI [OGSI]. OGSI defines the structure of notification ports and equally important Service Data Elements that can capture the state-change events that can be carried by the Notification service. The Notification service itself is being considered by the GGF OGSA working group with a preliminary specification available.

Many Grid projects have developed the equivalent of notification services but have not abstracted them as a clearly defined separate capability as for most of the current relatively modest projects, a sophisticated service is not necessary. We expect both the impetus from OGSA and the scaling up of projects will encourage more interest in this service. The JMS (Java Message Service) [JMS] based MyGrid notification service [myGrid-C] and NaradaBrokering system [NaradaBrokering] used in several Indiana Grid projects use publish-subscribe methodology which offers substantial flexibility with an overhead that is less than a

millisecond. There is often heated discussion as to whether notification should be "push" (as in simplest publish-subscribe) or "pull" from the clients. Probably future systems can offer both possibilities hiding the implementation in the invocation framework discussed in Sec. 8.2. XEvents [Slominski02A] is one of the best known application level event services which can be used in Notification systems. The EDG Relational-Grid Monitoring Architecture RGMA [EDGWP3RGMA] supports the natural producer-consumer (publish-subscribe) model used by notification services. However RGMA is usually considered as an alternative to MDS and is discussed in Sec. 7.6. The ambiguity between sections 7.5 and 7.6 and between the styles of messaging discussed above is seen in "the three ambiguous M's" – M can be messaging, metadata or monitoring. In this regard note that MDS ingeniously changed its name from Metacomputing Directory Service to more recently Monitoring and Discovery Service. Publish-subscribe systems effectively implement a (small) database needed to support the lookup of message properties needed in topic matching and so they can be used directly where a lightweight information service is needed.

Projects that have notification capability include Gateway [Haupt03A], the EDG European DataGrid Work Management with its logging and bookkeeping [EDGWP1-B] and the Globus GRAM monitoring which can be implemented as a service with either a Java CoG kit [Laszewski03A] or GridPort [Thomas03A] front-end.

7.6 Registry, Metadata and Information Services

7.6.1 Introduction

This section covers a rich suite of "look-up" services with the common characteristic that the data units being stored and queried are "small" in some sense and overlaps many areas that we have divided into registry, information aggregation, metadata (Semantic Grid) and provenance areas. As discussed above, one could in fact include notification events as another facet of this service. Naively these problems are just databases and we could perhaps stop the discussion and defer it to section 7.7 which describes the general Grid database support. However the critical nature and simple formats of the information to be managed suggests the need for special treatment. We suspect in fact that the hectic pace of the field and integration of many different approaches has produced too many separate solutions and we expect that a more unified approach using more professional database techniques will emerge as best practice. We also expect that all systems will come to support the federation of the basic metadata and registry services. This view suggests there will be more integration between the "small (meta)data" services in this section and the information Grid technology discussed in the next section. Metadata services will probably be implemented as optimized OGSA-DAI database services. We also expect important implementation issues connected with the OGSI Service Data Elements (SDE) supporting introspection on Web and Grid information ports. This suggests that some of the critical (meta)data will be stored in a highly distributed fashion within each service rather than in a modest number of centralized or even federated databases or registries. This model is already common in collaboration systems where "master" instances stream change data to collaborating clients; this is not a typical information query in that such a collaboration is set up as a peer group (session) and really this is a centralized information source but within a "small world". Use of SDE's raises the possibility that important data for "everybody" will be stored in individual services. Clearly one must develop a way of integrating this distributed SDE metadata with that in classic databases in a way that appears as a single lookup model. As discussed in section 7.6.4, we also expect the Semantic Web to significantly enrich the technology base in this area.

So we can expect significant implementation changes in the future and current systems and approaches such as use of LDAP are likely to change. However current systems are important as they define and explore needed capabilities.

At the lower level of the metadata hierarchy, we find information about the core Grid services which because of the (expected) OGSA compliance can be supported in a "Grid-wide" fashion. Their metadata either through direct compliance of interfaces or federation should be universal. This should be contrasted with more specialized application services whose metadata might be standard within a domain but not

broadly agreed or relevant. We can identify in more detail the following information types which can be included in this "small data" category.

a) Registry giving URL's in terms of URI's or GSR's in terms of GSH's.
b) From the Compute/File view, basic metadata specifying computers, users (historically what is needed to run Globus)
c) From the Web Services view, basic listings and simple look-up of available services
d) From the monitoring view, streams of information as to operation of Grid Resources
e) From the collaboration view, state changes in services to support coherence between object replicas and views
f) From the Semantic Grid view, sophisticated ontologies and metadata to provide intelligent lookup and matching of services
g) From each application field, specialized metadata to manage the data deluge
h) From the provenance and data curation view, metadata describing life-cycle and ownership of data

We now discuss these eight categories divided into four subsections even though it is not obvious that in the future a single technology might cover most of these areas. Categories a) b) and c) are in section 7.6.1; categories d) and e) in section 7.6.3; categories f) and g) in section 7.6.4 while h) is covered in final subsection.

7.6.2 Basic metadata including Registry

Here we cover the basic information systems from the current Grid (Globus) and Web service areas. The Globus MDS system [GlobusMDS] has been a critical technology and one of the first examples of a broad metadata service. Not surprisingly the implementation has some problems and we can expect it to be replaced by more powerful technologies. In the near term the European DataGrid has produced a version of MDS with a more powerful relational database core and their RGMA technology [EDGWP3RGMA] may become popular. MDS embodies both a particular LDAP "database" technology and a particular information structure. Unicore has developed a representation of jobs [Unicore-B]that is viewed as more sophisticated than that built into MDS and we can expect that both technology and data schema will continue to evolve. There is a joint US European activity GLUE [GLUESchema] to produce a common information schema but this appears to have a focus on existing Grids and it will need enhancements as Globus and Web service Grid converge.

Turning to the Web service field, the starting point is different. We are trying to allow users and more importantly other services find out which services are available and how one should access them. There is no special emphasis on the Compute/File Grid schema supported by MDS. The two basic technologies are WSIL (Web Service Inspection Language) [WSIL] and UDDI (Universal Description, Discovery and Integration) [UDDI-B] [UDDI4J] [UDDI-A] but these have been found insufficient by most Grid and indeed Web service applications (see appendix, section A.2.6.1.1); we already noted this in section 7.4 [OGCWS]. The basic problem is that these two technologies store essentially no useful semantic information about the Grid or Web services and so it is very hard to ensure that one is choosing a service with both the correct functionality and with matching interfaces. Note that MDS and RGMA have substantial semantic information but it is largely focused on the information needed to run jobs on computers. It is not clear if UDDI and WSIL will evolve or rather new approaches such as those discussed in section 7.6.4 will replace them. The future information systems needed for Web services will go beyond the generic description frameworks of UDDI or WSIL to incorporate the rich ontology support available in the Semantic Web [SemanticWeb]. Essentially, two of the key features of information services for Grid and Web Services are knowledge representation and intelligent search (search by meaning as derived from ontologies, for example). These two areas are active areas of research in the Semantic Web covered in section 7.6.4. In the medium future, we expect "many flowers to flourish" and the different approaches to evolve independently until consensus develops and allows the current systems to merge into more powerful and general approaches.

There is a common "registry function" in the current MDS and Web service technologies which naively can be thought of as generalizing the familiar DNS. This needs to be implemented in a robust fashion and avoid hard wiring as much as possible. Relevant technologies here include Jini [Jini], the IEEE Service

Location Protocol (SLP) [SLP], and the Service Discovery Service [NinjaSDS] from Berkeley. All of these approaches rely on the use of multicast requests to discover a registry service(s), and therefore are often restricted to a limited part of the network. Jini provides a useful approach for enabling Java classes to expose and publish their interfaces with a "look-up" service, along with proxy objects that enable subsequent invocation of the Java classes. The proxy object needs to be downloaded by the client to initiate a session, and can thereby enable a number of different protocols to co-exist. The approach relies on the assumption that all services must define their interfaces using the Java type system, and also that subsequent search for suitable services in the registry (the lookup service) is restricted to these data types. Jini provides the notion of "leasing" (or soft state), which is also now available in OGSA. The Service Location Protocol uses a series of filter expressions to identify the location, types and attributes of a service within a part of the network. The query language is simpler than Jini's. The Service Discovery Service also uses the same principles as SLP and Jini, and support a simple XML-based exact matching query type. The particular advantage offered by SDS is the use of secure channels to discover and publish service metadata. Furthermore, SDS servers can be organised into a hierarchy, enabling an administrator to analyse (and subsequently improve) performance at particular points in the hierarchy. Each SDS node in the hierarchy can hold an aggregated index of the content of its sub-tree.

There are many other projects in this area for "all distributed systems" must have such a low-level registry and meta-data capability. Hoschek has prototyped a peer-to-peer technology for Web Service semantic service discovery [Hoschek03-A]. Note that many approaches such as Gateway in [Haupt03A] do not separately "package" Registry functionality but support it internally to the system. NeesGrid is implementing a new basic metadata system [NeesGrid-B].

7.6.3 Information Aggregation Web Services

There are important classes of Grid and web services to and from which information is streamed on a continuous basis. Examples include performance or monitoring services; network status measured in the different ways described in section 7.11; computer status with a catalog of individual jobs; service management frameworks logging resource availability and performance; performance monitoring of running applications; collaborative environments [Fox03C] where one instantiation of a service streams out changes either for its internal state (shared event model of collaboration) or at the simplest for its rendered display (as used in shared display approach to collaboration). This area is best exemplified by the well known HotPage [HotPage] technology from SDSC which is used to display the job status on HPC resources; this has been incorporated in the more advanced GridPort system from the University of Texas [Thomas03A]. The latter has enhanced HotPage to the more powerful GPIR GridPort Information Repository [GPIR], which can handle job, NWS and MDS data. This whole area can be called "information aggregation" and we expect the different applications to be implemented as Web services which as discussed in section 7.5, are closely related to the notification area.

Plale at Indiana University [PlaleWS] has also developed a portal interface and service for monitoring the performance of many different information system implementations (LDAP-based systems, relational databases, and native XML databases (see notification). We expect this area to grow in interest as more attention goes into performance and monitoring areas that are essential to a high performance robust Grid.

7.6.4 Semantically rich Services and Ontologies

It seems certain that there will be dramatic growth in the area of metadata associated with both web services and the Grid. As discussed in section 7.6.2, current low level systems are not easily generalized to properly address the need for "metadata rich Web services" which in section 5 we suggested was a key hallmark of (future) Grids. There are two major thrusts we can identify; firstly we have many existing domain specific metadata repositories and secondly there is the technologically based Semantic Grid initiative. We will discuss these thrusts separately and note first that many domain specific metadata catalogs are maintained "outside the Web and Grid community" by particular application areas and so cannot be easily changed. Thus such catalogs are best approached by first wrapping them using OGSA-DAI technology (section 7.7) and then linking this wrapped database to a general metadata service offering a Semantic Grid like capability. The Storage Resource Broker from SDSC with its associated metadata

catalog MCAT [SRBMCAT] has a Web service interface which is a good example of this approach for the many repositories its supports. Bioinformatics illustrated by the European Bioinformatics Institute [EBI] has for many tears supported such critical domain specific data and meta-data catalogs. Again Daresbury Laboratory puts substantial effort into supporting metadata for many fields and has developed both a common framework [Matthews01A] and portal interface [CCLRCeSCDP-A]. We expect the Semantic Grid activities [SemanticGrid] described below will be important in this task of integrating and enhancing application specific metadata. In fact "new" application specific catalogs should adopt Semantic Grid ideas as their core technology and it would be important to test this out in some particular cases. Currently this powerful metadata technology is being used for the "new problem" of adding semantic richness to Web services and we now discuss this.

From the Grid and Web Service arena, WSDL [WSDL] provides a standard description of the formal interfaces (number and nature of parameters) and increasing number of activities like OGSA will only increase the number of services with defined interfaces. However knowing the formalities of the interfaces does not imply much about the meaning of the service and whether two interfaces match in detail. Thus there is general agreement that this is a critical area for the web and the highly visible Semantic Web activity [SemanticWeb] has a current focus on adding semantic richness to Web service interfaces. The Semantic Web has overlapping areas, with RDF [RDF-A] serving as a Web standard for metadata descriptions. Much richer ontology languages—RDF Schema [RDF-B], DAML OIL[Fensel00A] [Fensel00A] [Hendler00A] and its follow-on, OWL [OWL], are also being developed to supplement RDF to provide an extensible, object-oriented framework for building metadata classification and taxonomy systems. UK e-Science projects such the Bioinformatics Grids, MyGrid [myGrid-A] and DiscoveryNet [DiscoveryNet-A] are pioneering these applications of Semantic web technologies to Bioinformatics Grid applications. Indiana University is starting such a project in Earth Science [SERVOGrid]. There is currently little experience in key tools; those to develop and test ontologies and those to add, store and search metadata. This will be a critical activity in the near term. Note that as discussed in section 7.4, ontologies provide one approach to programming all or part of the workflow problem and it will be interesting to see what is best described in this way and what should be expressed by conventional programming.

Southampton has produced a general toolkit SDT [SDT] to support rich metadata catalogs. It will be important to look at this and similar new systems [NeesGrid-B] and understand what type of support the different metadata categories described in section 7.6.1 need.

7.6.5 Provenance

Provenance, sometimes known as lineage, is a record of the origin and history of data items. It can be thought of as an 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 database. It is familiar operationally in experimental science where one's Lab book provides such a record. This can be generalized to e-Science where any user of data needs to know how it was produced. Although clearly needed, the methods for recording provenance are not well codified and the Semantic Grid offers an important technology for specifying such a data lineage. Support for provenance is an essential requirement in an e-Science environment as data sharing is central to the basic concept of a virtual organization. Provenance is key to establishing the quality, reliability, and value of data in the discovery process. We expect a growing number of activities in this area with MyGrid already looking into this. [myGrid-D]

7.7 Information Grid Services

Although Information Grids are perhaps the most important class of Grid problems, there is so far rather little work on generic services. The first examples in this area come from the astronomical virtual observatories [NVO] [iVOA] with a good description of the architecture and metadata given by Williams [Williams03A]. The US Virtual Observatory uses powerful Microsoft SkyServer technology [Skyserver] but this is not directly generalized to other applications; further the current deployed systems have excellent capability but are not structured as Web or Grid services. The latter is certainly a goal of the UK e-Science AstroGrid project [AstroGrid]. Virtual Observatories can be generalized to other fields including Earth Science [SERVOGrid] where the observation comes from satellites, seismic sensors and the like. General

issues in Information Grids are discussed by Watson [Watson03A] for databases and Kunszt and Guy for the type of large mainly file-based data archives needed at CERN [Kunszt03A].

As mentioned above, the SDSC Storage Resource Broker Technology is an example (see appendix section A.2.7.2.3) of a Web Service interface to a largely file-based data repository [SRBMCAT] [Moore03A]. NeesGrid is building a data repository Grid Service [NeesGrid-B]. Spitfire from the European DataGrid was one of the earliest examples of a Grid interface to a general relational database [Spitfire]. This has been generalized by the e-Science OGSA-DAI project (appendix sections A.2.7.1 and A.2.7.3.2 [OGSA-DAI]) to provide a Grid interface to the full range of databases including relational and XML structures. This project will in the next 2 years [OGSA-DAIT] provide additional pure Web service interfaces and cover the full range of database structures. OGSA-DAI will also address the important database federation issue (termed the "distributed query" problem) as although this can be considered as "just a database" problem, Grids seem the natural approach to any kind of federation for this is how one builds virtual organizations. We have already discussed in sections 6 and 7.1, the need for hybrid Grids linking the Information and Compute/File styles. This involves workflow for as shown in fig. 6.3, one must integrate filter services with data services and this will be addressed by the next round of OGSA-DAI activities. Such integration is critical in many applications such as Bioinformatics, Environmental Science and the Virtual Observatories. We can also expect important advances from integrating the ideas of peer-to-peer networks with Information Grids [Fox03C] [Hoschek03-A] as we know the great popularity of the many different P2P file sharing systems such as Kazaa [Kazaa]. Possibly one could build a lightweight JXTA Grid [JXTA] supporting such a P2P Information Grid model.

Another type of Information Grids is built around data from sensors and instruments rather than the high-level databases. Such sensors could be accelerators [Laszewski02A] [Johnston03B], X-ray crystallography machines [Krishnan01A] or chemical laboratory devices as in the combinatorial chemistry Comb-e-Chem project [CombeChem]. Another example from the DAME e-Science project [DAME] involves gigabytes of data per day streaming from aircraft engines. Standards do exist in this field such as the new Space Link Extension SLE standard [CCSDS-A] from the international space community. This dictates how ground-based clients will interact with space sensors but unfortunately it was designed before Web and Grid services were developed. Thus this standard would need to be modified to add in particular a multi-tier rather than the current client server architecture. This is an example of a general problem in working with outside standards and deployment bodies. Many have ongoing and highly effective activities developed with very different underlying principles from Information Grids. We will have to learn how to integrate these with a service architecture and WSDL style interfaces.

Another important area for e-Science is data curation; essentially how one uses the techniques and goals of digital libraries and archives with the data deluge of electronic information [Macdonald02A] [Curation-A]. Some experience exists with electronic publication of articles but little for other information forms. The field of Bioinformatics has substantial successful experience with databases subject to formal curation [EBI] but some other fields simply do not tackle this at all. We expect it will be important to start several pilot projects and develop the needed generic supporting Grid services. Much progress can be expected and indeed is essential in this area over the next few years.

7.8 Compute/File Grid Services

Compute/File Grids are the best developed facet of Grid technology and impressive well tested technologies exist in this sector. However partly because of this, the field is not easy to discuss. The systems have rich overlapping capabilities and all will presumably evolve to become true Grid services with WSDL interfaces. We show schematically in fig. 7.2, the ten sub-categories we have chosen for classification of services in this area. As usual this is not unique and most technologies have capabilities in several of the sub-categories.
There are three major efforts driving a lot of the software in this area. These are the Globus project [Globus-A], Condor [Thain03A] and the European DataGrid EDG [EDG-A] which is building several important capabilities off Globus and Condor. Useful links are the different work packages [EDGWP] of the EDG which correspond to different capabilities discussed in this section; [EDG-B] gives status of EDG software releases. In addition, the US has two largish software deployment projects funded by NSF; the

Middleware initiative [NMI] and the Virtual Data Toolkit VDT [VDT]. The latter is part of the Griphyn [GriPhyn] project whose research focus is virtual data (section 7.8.8). However the VDT includes robust versions of many useful core Compute/File services.

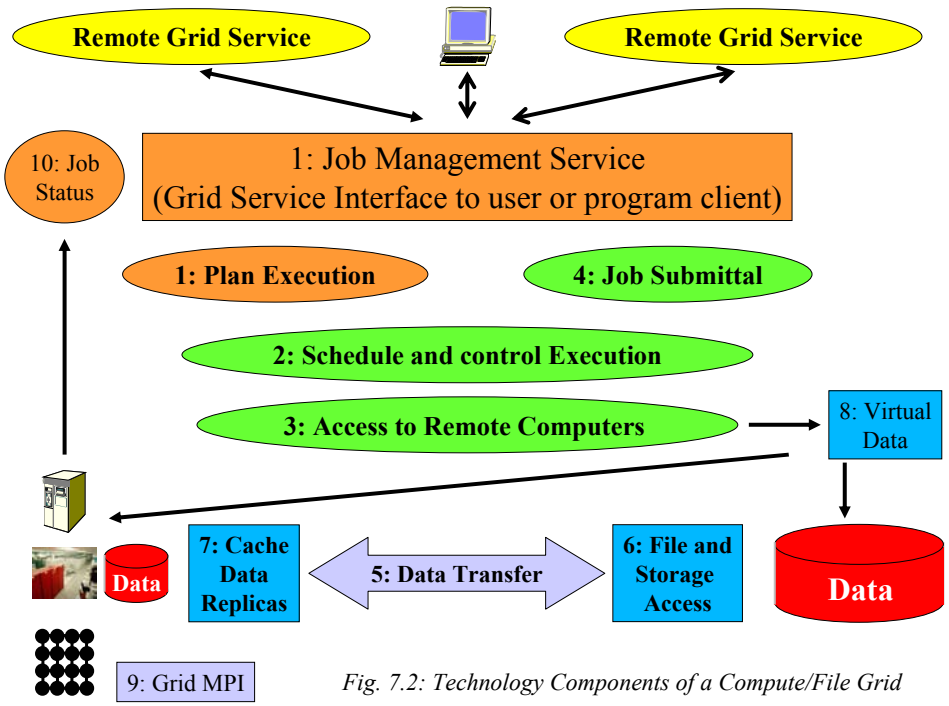


Fig. 7.2: Technology Components of a Compute/File Grid

There are several other important activities in this area. The Alien Grid [AlienGrid] uses a Perl infrastructure and builds on Condor ClassAds EDG and Globus for a lightweight Grid for the CERN ALICE experiment [ALICELHC]; it is being applied to other Grid applications. GridSystems is a successful commercial vendor [GridSystems] of Compute/File Grids in the utility computing area.

7.8.1 Planning and Management

In the first category, we include the overall management of the execution of a job. We also put the planning activity here although as suggested in fig. 7.2, this could be separated off. Complete systems like Condor [Condor] and Unicore [Erwin02A] include planning and management functions although one would often build additional functionality in many portal systems that would use Condor, Globus [Globus-A] or other scheduling systems (sections 7.8.2 and 7.8.3) at lower levels. One example is the Gateway or Mississippi portals [Haupt03A], whose sophisticated management capability interfaces in this fashion. In particular, it is this level that must present a service (WSDL) view to users or other services. Thus systems like the Java CoG kit [Laszewski03A], the GPDK development kit [Novotny03A] or GridPort [Thomas03A] are used to build management systems. The EDG in work package 1 [EDGWP1-B] has a sophisticated resource broker at this level [EDGWP1Arch]. NCSA is developing an application management system for several of their applications [NCSAGAMS].

Planning requires the ability to match jobs and computers and the ClassAds system from Condor [Condor-ClassAds] is popular core technology here; it has a flexible way of specifying both jobs and computers so that good matches are possible. Note that planning is a sort of "virtual version" of the scheduling described in section 7.8.2. In general planning needs to include both workflow and "virtual data" (sections 7.4 and 7.8.8) and as these ideas develop, we can expect the planning systems to evolve. Globus has a recent capability Pegasus [GlobusPegasus] in this arena.

7.8.2 Scheduling

All large computing platforms need scheduling capabilities to manage and optimize execution with a mix of real-time and batch queues. Popular technologies include PBS (Portable Batch Scheduler [PBS-A] [PBS-B], Sun Grid engine SGE [SGE] and LSF from Platform Computing [Platform]. These are mature technologies as is the Condor system from Wisconsin which has functionalities in many key parts of section 7.8. There are overlaps between these schedulers and the Desktop Grids (cycle scavengers on "idle workstations") like Entropia [Chien03A] discussed in section 7.1. The Alien Grid uses a pull (to resource) rather than push (from broker) approach for relation between resource broker and resources themselves.

We can expect continued improvements here with perhaps the most important long term changes being in the development of scheduling of services and workflow as opposed to jobs.

7.8.3 Access to Remote Computers

As in section 7.8.1, some portals offer "seamless access" to a variety of computers. If you just need to cover a few machines, then this can be arranged "by-hand" by coding the explicit submission scripts for each computer of interest. This is a common, successful and effective when one can identify the systems to which access is needed. One example of this is the practical supercomputing toolkit [PST], which offers a uniform interface to the major machines used by the US DoD high performance computing community. Globus offers the powerful concept of a common interface to multiple machines through the Globus Resource Allocation Manager GRAM [GlobusGRAM] controlled by metadata discussed earlier in section 7.7.2 with the MDS service. GRAM interacts with Condor in distinct fashions. Condor-G [CondorG] can run "underneath" Globus and allows the GRAM to submit jobs to pools of Computers managed by Condor. Alternately the Condor Glidein [CondorGlideIn] allows GRAM controlled computers to be accessed from Condor. Unicore [Erwin02A] [Unicore-A] [Unicore-C] is a important "seamless access" technology having many best practice technologies which have been discussed under the security, workflow and metadata categories. The European Union funded GRIP (Grid Interoperability) project aims to integrate Globus and Unicore systems.

The near future will see capabilities here packaged as Grid services with for instance GT3 evolving GRAM in this direction.

7.8.4 Managing Job Submission

As well as technologies to provide access multiple computers through a single interface, one must be able to submit and manage the job submission itself to individual systems. There are many portals that control specific computers and of course the basic Grid technologies Condor, Globus and Unicore offer this capability. As OGSA standards are developed for jobs and computers, one should see a uniform approach to this with each computer being a Grid service. The difficult security issues will still make it hard to offer seamless access to multiple machines even with a standard interface.

7.8.5 Compute/File Grid Shell: Data Transfer

Data transfer is an important area as it underlying all of distributed computing. GridFTP (building on earlier systems such as GSIFTP) from Globus [GlobusGridFTP] offers OGSA compliance, high performance and compatibility with Globus security. We can expect further improvements in areas of fault tolerance, compatibility with firewalls and Grid service packaging. Related technologies include the GSI-OpenSSH [GSIOpenSSH] from the NSF Middleware Initiative and the fault tolerant Shell [FTShell] from the Virtual Data Toolkit. Note as discussed in section 7.9.1, a basic Grid Shell underlies Compute/File Grids for both Globus and Legion [Natrajan02A]

7.8.6 File and Storage Access

The interaction between compute jobs and typically file based data is critical to the compute/file class of Grids. This involves two types of technology discussed in this and the following section. Here we describe the "wrappers" that allow convenient remote access to data. There is GASS or Global Access to Secondary Storage from Globus [GlobusGASS] and the Storage Resource Management SRM [SRM] which is

collaboration between the European Data Grid (work packages 2 and 5) and several US Department of Energy laboratories. The latter involvement reflects the importance of this technology in particle physics applications [PPDG]. EDG work package 5 has a more general Storage Element wrapper [EDGWP5] including SRM support. Other projects in this area include the Web service wrapping of such capability from the Jefferson Laboratory [Watson03B]

7.8.7 Replica Management and Caching Technology

The second area of importance in handling of files and more generally data repositories involves the distributed caching [EDGWP2-B] of information across large distance networks. There are major efforts in this area from both Globus and the European DataGrid with again particle physics as a major driver. GDMP Grid Data Mirroring Package [GDMP] based on Globus is the current approach developed by the Particle Physics Data Grid PPDG [PPDG] and EDG. EDG has new technology including replica (cache) management [EDGWP2RMS] and a locater RLS and RMC catalog service [EDGWP2RLS]. This includes their Optor or ROS Replica Optimization Service. Globus has a Replica Management [GlobusReplMan] and Catalog [GlobusReplCat] Service.

7.8.8 Virtual Data

Virtual data refers to the interesting concept that data can be referenced indirectly through the workflow needed to create it. This is the research focus of the GriPhyn project [GriPhyn] with their Chimera [Chimera] system distributed in the Virtual Data Toolkit [VDT]. This area is still at a fledgling stage but is considered important by the particle physics community as discussed in European Data Grid requirements [EDGWP8HEPCAL].

7.8.9 Parallel Computing

Parallel Computing Systems can be accessed as a particular computer using the technologies described earlier in this section. However one can also take message passing jobs and run them "over the Grid" although this will only give good results for applications that can tolerant the large 1 to several 100 millisecond latency of network connections. MPICH-G2 [MPICH-G2] (made available through the NSF Middleware Initiative) and PACX-MPI [Mueller02A] are two Grid-enabled MPI implementations.

7.8.10 Job Status

Job status capability is critical in Compute/File Grids and has been discussed in detail under notification (section 7.5) and information aggregation (section 7.6.3).

7.9 Other Grid Services

Here we discuss some interesting services that often are generally useful for several types of Grid and so were not discussed in the previous two sections on Information and Compute/File Grids.

7.9.1 Grid Shell

The concept of a Grid Shell can be discussed at two levels. As remarked in section 7.8.5, a generalization of UNIX Shell commands underlies both Globus and Legion systems supporting job and file related capabilities. However the concept of a set of "atomic" commands embodied in the Shell can be generalized to a set of important Grid services of wide-ranging applicability. Bill Joy exploited this in the peer-to-peer system JXTA [JXTA] whose Shell supports collaborative functions. Thus as well as referring to the Grid enhanced UNIX commands, we can use the shell terminology to describe a suite of Grid services. Further in analogy to UNIX, one can develop a general Grid Service interface equivalent to *sh* in that it can invoke "any" other Service. This latter idea has been superficially explored at the Grid Computing Environments working group at GGF and at Indiana University with a prototype implementation [Fox03D] [Nacar03A].

7.9.2 Accounting and Grid Economies

Very important areas are accounting and the economics of Grid computing. Accounting itself is closely linked to security issues as suggested by the AAA (Authentication, Authorization and Accounting)

category used in the Grid Forum and other places. More exciting but not necessarily more important is the idea that the Grid suggests new approaches to charging and accounting for the use of resources. This idea has been extensively explored by the peer-to-peer community [Oram01A] with novel concepts of bartering, "the tragedy of the commons" (what happens if a resource is totally free) and digital cash. Computational economies or markets are the typical way such ideas are presented. Some of the pioneers in this field include the work at Santa Barbara [Wolski03A] and the Universities in Melbourne Australia [GridBus] [EconomyGrid] [Buyya02A]. A new effort spanning both AAA and Computational economies has just been started as part of the UK e-Science program (see appendix sections A.2.9.1.1 and A.2.9.1.6) [UKeSMarket].

Logically distinct but operationally related are Grid technologies to support "parametric modeling" or the control of multiple independent but related jobs. In a typical case one explores a parameter space by launching several jobs with a single replicated code but distinct input parameters. Nimrod-G [Nimrod**]** and APST [Casanova03A] are well known tools in this area. Desktop Grids and Condor are related technologies.

7.9.3 Fabric Management

A seemingly important area is fabric management which means systematically controlling and monitoring the system hardware and software of Grid resources. The sensitivity of application software (such as that of the European DataGrid) to particular versions of Linux and other operating systems and tools emphasizes the importance of this area. LCFG [LCFG] was originally developed by the Informatics school at the University of Edinburgh to manage the computers in their department. However it has been successfully extended to larger scale application involving the core software installations on both clusters and Grids. Curiously there appears no such software in common use in the US but there are two efforts to extend LCFG in Europe. In the near term, work package 4 of the European DataGrid has developed with Edinburgh an enhanced next generation LCFG(ng) [EDGWP4LCFG]. On a longer term view and with attention to research issues, Edinburgh has teamed with the Hewlett Packard laboratory in Bristol in the UK e-Science GridWeaver project [GridWeaver]. This builds on LCFG and the HP SmartFrog system supporting service specification and deployment [SmartFrog] for utility computing; see appendix section A.2.9.2.1.

7.9.4 Visualization Datamining and Computational Steering

It is clear that data analysis and in particular visualization are important Grid services. Although scientific visualization is critical and has been pursued intensely, there is no obvious consensus architecture and correspondingly it is hard to design general Grid services to support this field. The well known Utah group which developed SCIRun is packaging their visualization system as a Grid service as part of the NCSA Alliance portal activity [NCSAGAMS]. A UK e-Science workshop [UKeSSDMIV] reviewed both visualization and the related datamining issues. GADS [GADS] is addressing some of these issues in environmental science.

One reason for real-time visualization and analysis is to support computational steering; adapting in real-time the execution of a program based on its initial results. Such a capability is naturally thought of as a service (related a little to debugging) where the user is presented with a portal displaying both the current results and the ability to change parameters defining the execution of a remote job. The work of Parashar's group at Rutgers pioneered this in the DISCOVER project [Mann03A] which had a Grid service structure implemented in CORBA. The UK e-Science RealityGrid (appendix section A.2.9.1.3 and [RealityGrid]) project is a major new effort aimed at controlling material science simulations via Grid-enabled computational steering.

7.9.5 Collaboration

We have already discussed collaboration in terms of notification (section 7.5) and information aggregation (section 7.6.3) services. These provide the core technology to support the object sharing that is the heart of collaboration. Collaborative systems support three types of capability

a) Audio-video conferencing

b) Common tools like whiteboards, text chat and instant messenger
c) More general shared applications such as PowerPoint or perhaps a scientific visualization.

Further there is a control system that sets the details of a collaboration such as which clients and applications are involved. In a Grid or web service architecture, one can implement this elegantly [Fox03C] with a web service for the control system and each shared capability constructed as individual or replicated web service where one shares either the input (replicated) or output (single instance with shared view) ports.

Best practice for collaboration in the Grid arena is the Access Grid which is being repackaged using Globus and Grid service technologies as part of the SWOF Scientific Workspaces of the Future project [SWOF]. This focuses on high-end videoconferencing (category a) above) and can be linked to commodity technology through VRVS from Caltech [VRVS] which is popular especially in the particle physics community. CHEF [CHEFportal] provides excellent implementations of some of the tools in category b) and is being developed as part of the NeesGrid [NeesGrid-A]. CoAKTinG [CoAKTinG]is a UK project examining enhancements to Access Grid based collaboration. Indiana University has developed a Web service approach [Uyar03A] to cover all aspects of collaboration. The XGSP (XML General Session Protocol) system links the Access Grid, SIP and H.323 approaches with a common controller that using NaradaBrokering [NaradaBrokering] as a distributed messaging broker can scale to very many simultaneous users. This can integrate with Polycom [Polycom] commercial H.323 video conferencing and WebEx [WebEx] style collaborative applications. However this system is still only a research prototype.

7.9.6 Packaging

The Grid tools need to be packaged to allow robust convenient deployment. One can use technologies like RPM [RPM] familiar from the Linux community but specialized systems have been developed for Grid software. The NSF middleware initiative uses Gridconfig to manage the configuration of their software components [Gridconfig] and GPT (Grid Packaging Tool) [GPT] for packaging. Pacman [Pacman-A] from the high energy physics community is used by the VDT [Pacman-B].

7.9.7 Other Technologies

There are of course many other tools needed and available to support Grid applications and these can be thought of as parts of the emerging Grid Shell introduced in section 7.9.1. Two examples developed by the UK particle physics GridPP project are SlashGrid [SlashGrid] which is a framework for Grid aware file systems (see section 7.8.6) and GridSite [GridSite]. The latter is a Web site content management system which is analogous to the Apache Slide system [Slide]. The latter supports the IETF WebDAV distributed authoring and versioning standard [WebDAV]. Discussion of these and other technologies can be found in the appendix, sections A.2.9.1.5, A.2.9.2.3 and A.2.9.2.4.

7.10 Portal Services and Problem Solving Environments

This topic could cover a broad range of topics but here we will focus on so-called Portal Services which are the Grid components that control the marshalling of information from a variety of Web or Grid services and allow the user to view and interact with them. This problem is not just building a Web page as one needs to present the views of multiple Grid services in a way that is easy to customize for users and administrators. The previous sub-sections of this section have covered the services themselves with section 7.4 covering the important integration of multiple services needed both in Grid Computing Environments and Problem Solving Environments. The reviews in [Fox03A] and [Fox03B] describe both the middle tier and user interface integration issues and reference over 50 papers in these areas.

The service architecture with its refinements described especially in Section 7.2 implies a component model for the middle tier and the underlying idea of modern portals is to match this with a component model for the user interface which will be built from “document fragments” with in the simplest case, one fragment for each service. The portal then integrates or more precisely aggregates the individual fragments into a web page.

Fig. 7.3 shows this key architectural idea. We assume that all material presented to the user originates from a Web service which is called here a content provider. This content could come from a simulation, data repository or stream from an instrument. Each such Web Service has resource or service facing ports (RFIO in fig. 7.3), which are those used to communicate with other services. Here we are more concerned with the user-facing ports (shown on the right of the Web service) which produce content for the user and accept input from the client devices. These user-facing ports use an extension of WSDL, which is being standardized by the OASIS organization. This is called WSRP or Web Services for Remote Portlets.

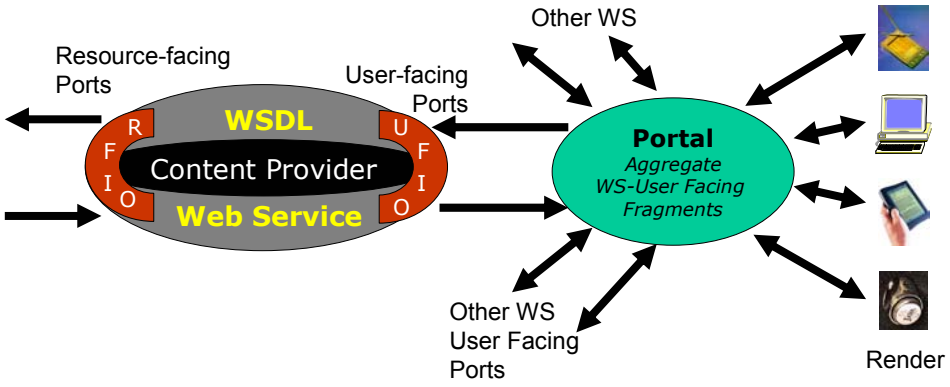


Fig. 7.3: Portal providing aggregation service for document fragments produced by user-facing ports of a Content providing Web Service

[WSRP]
Most user-interfaces need information from more than one content provider. For example, a computing portal could feature separate panels for job-submittal, job status, visualization and other services. Figure 7.4 illustrates these ideas with a portal being developed for the NCSA Alliance [OGCE]. One sees 4 separate interfaces (3 on left and one on right) to different GCE Web services. Further capabilities are aggregated using tabs at the top.
One could integrate this in a custom application-specific Web service but it is attractive to provide a generic aggregation service. This allows the user and/or administrator to choose which content providers to display and what portion of the display real estate they will occupy. In this model each content provider defines its own "user-facing document fragment" which is integrated by a portal. Such aggregating portals are provided by the major computer vendors and also by Apache in its well known Jetspeed project [Jetspeed].
Portlets provide the desired component model for user interfaces in the same way that Web Services represent a middleware component model. Using this approach has obvious advantages of re-usability and modularity. One then has an elegant view with workflow integrating components (Web services representing nuggets) in the middle tier and aggregating portals integrating them for the user interface.

We expect the WSRP user-facing port interface to be supported by a growing number of portal systems. As well as commercial products like WebSphere and open source Jetspeed, there is a GridLab project Gridsphere [GridSphere] which is prototyping Grid enhancements to this architecture.

The portlet approach has an attractive model for Grid system development. It requires that service developers build both individual Web/Grid service functionality and user interface together as components that can be assembled in the middle tier and client respectively. Use of WSDL, OGSA, and other metadata standards together with WSRP ensure interoperability and re-usability both in the middle tier and on the client. This has been found an excellent approach for distributed teams with relatively easy integration of the contributions of different teams.

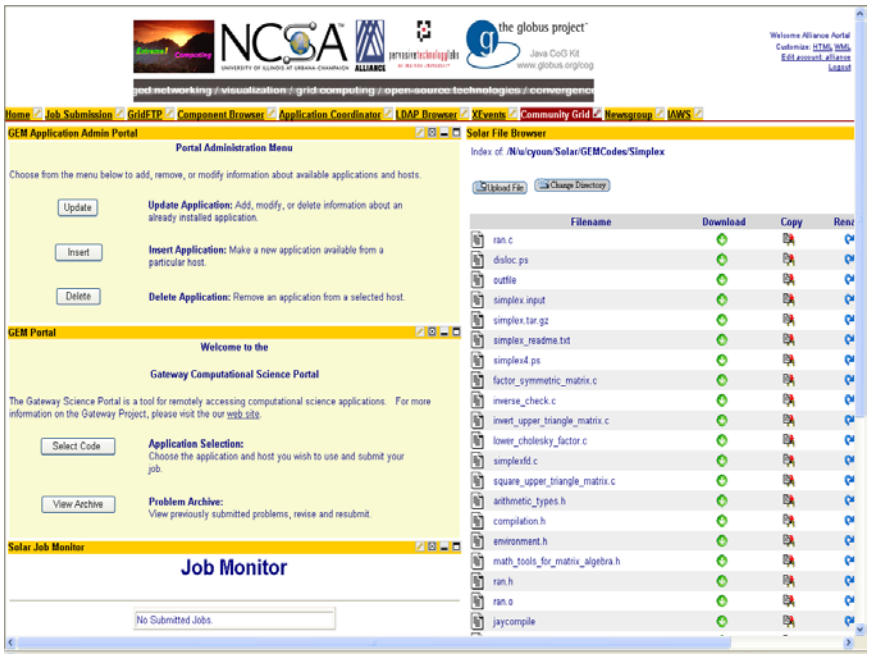


Fig. 7.4: Example of a Jetspeed-based Portal with aggregation of interfaces to several computing services

7.11 Network Services

More details on the interaction of Grids and networks can be found in section 8.11 and the appendix section A.2.11.1.

7.11.1 Network Performance, Monitoring and Information Systems

There are currently a large number of disjoint activities [NetworkMonitor] in the sector of network performance and characteristic monitoring for the Grid, each serving a particular purpose. A representative, but not exhaustive list includes in the EU the work done under the DataGrid EDG project (major site-to-site monitoring and publication to MDS and R-GMA) [EDGWP3IMS], the UK e-Science monitoring infrastructure (from Daresbury Laboratory [UKeSGridMon]), aggregate traffic statistics (available on an ad hoc basis form core providers) and the recent Geant/TERENA [Terena-A] initiatives [Terena-B] in core performance monitoring and diagnostic authority chains. In the US there are many relevant projects such as the well known Network Weather Service NWS [NWS], the SLAC based IPPM [IPPM] system and the Internet-2 end-2-end performance initiative [Internet2e2epi]. All existing network monitoring schemes tend to use the same well known measurement engines under the hood (PingER [Matthews00A], IPERF [Iperf], UDP throughput, GPS based one-way delay, FTP throughput), but each implement a context-specific framework for carrying out measurements and tend to be fronted by a context-specific visualisation front end, and all speak different languages.

Each of these has been very successful in the context which it has been created, but as a general statement none of these are designed to be "joined up" nor are they cast in a way which easily admits of using them as a core set of heterogeneous low level services to provide information into different higher level views (i.e. resource scheduling, Grid operations tools, SLA monitoring….). To put this succinctly one can easily today measure end-to-end TCP throughput between two arbitrary sites but this is in general not done a scalable way and one cannot make use of network information along the path to diagnose why it might be poor or build up performance information in a scalable way.

7.11.2 Network Services and Operations

In this sector of "other than best efforts IP", then simply put there are no such services available at this time. The only candidate "better than best efforts" service being rolled out is diffserv (differentiated services) IP premium, which is still not available end to end within Europe, and not at all in the US. In contrast the complementary service known as "less than best efforts" is configured in many domains in the EU and US and has been successfully demonstrated in one off end-to-end tests. Other more sophisticated services like point-to-point Ethernet circuits still exist only in research projects.

7.11.3 Network Reservation

Commensurate with this we see that the current efforts in the control plane software necessary to give Grid middleware access to these services are all limited to sub sectors of the phase space. There are various EU Framework-5 projects dealing with control plane software within a limited domain, but these do not have a Grid middleware viewpoint (they concentrate more on bulk carrier viewpoints). Coming from the other end the well known GARA (Globus Architecture for Reservation and Allocation) initiative has been around for some time which covers many aspects of the phase space, but doesn't itself deal with the detailed scalable AAA (Authentication, Authorization and Accounting) system (it provides hooks for this). GARA is certainly a candidate for "case hardening". There is also an initiative within DataTAG [DataTAG] and the "lightpath optical switching" community to develop [GommansWS] a workable AAA infrastructure to allow requests for resources to be made at ingress, egress and all points along a network path. There some efforts going on into deploying both of these for proof of principle demonstrations.

Thus in short the current situation is that we are at the very beginning of seeing the availability of novel services technically on the wide area network, and we are similarly at the beginning of seeing some of the control plane software necessary to give access to grid middleware.

Part III: Gap Analysis

8 Grid Technology Gaps

- Need to distinguish success of project in its own world from generating infrastructure and methodology that can be generalized to other e-Science activities in same or other disciplines
- We assume a service model and in greater detail that Web Services will be basis of e-Science infrastructure. The service model is consistent with traditional distributed object models like CORBA or Java and it is not hard to migrate from these object models to Web Services.
- Listed gaps may be inconsistent and "wrong" as correspond to individual views; we need to "weight" with number of people expressing
- Note many of gaps correspond to UK wide deployment of capabilities now being researched or developed in existing projects

8.1 Different Types of Grids and their Federation

8.1.1 Introduction

The general ideas of e-Science or the Grid encompass a very heterogeneous set of possibilities differing in both functionality and implementation. For example in e-Science we need to consider the LHC Grid of ATLAS and CMS experimenters with thousands of scientists and hundreds of petabytes of data. At the other end of the scale consider the training Grid set up for teachers and students of some course; maybe 20 people and a few computers and gigabytes of data. In business, a major corporation like IBM would have an internal Grid supporting a few hundred thousand people with large scale resources and software systems. A medium-sized business like Capitol Radio (section 4.3) might prefer a peer-to-peer structure linking their roughly 15 sites in a flexible fashion. In the military case the infrastructure to support storage and analysis of imagery from satellites would be a large scale Grid rivalling that of IBM or LHC. In contrast a reconnaissance mission of just one or a few vehicles would be supported by a dynamic small scale Grid with very different requirements in terms of number and nature of devices and services. Another feature can be seen in the US defence program called "Global Situational Awareness" whose aim is to "monitor anywhere anytime" with a network of sensors, analysis stations and analysts. This is naturally architected as a Grid but has the constraint that we can't afford to build new weapon systems; rather we must evolve and integrate existing systems. Here one approach is to take each existing system and re-architect each as a Grid; then the total DoD environment must be built by federating these existing Grids.

As well as differing in size and requirements, Grids will differ in terms of base technology used. This technology could have been chosen either because it seemed to match the requirements well or for some other more subjective reason. Already Grids have been constructed from Globus, Avaki, .NET, Jini, and JXTA technologies or just by integrating Web Services.

In e-Science we see clearly defined Information and Compute/File Grids in the language of Section 6. There are also composite Grids integrating both functions as in areas like Earth and Environmental Science. These need to integrate large databases with massively parallel simulations.

There were many comments on the need for fault tolerance and this motivates the Autonomic Grids; the need for this large is clear in particle physics to manage in a scaling fashion the analysis of many petabytes of data per year.

The challenge of dealing with the heterogeneity in requirements and technology is of critical importance. Large enterprise Grids must have appropriate robustness and security; the small Grids need very dynamic

easy to deploy technology which is typified by peer-to-peer systems. In section 8.1.7, we discuss Grid federation as a way of linking different Grid islands together.

8.1.2 Summary of Gaps

- Need lightweight (measured in both deployment time and complexity of software needed) Grid infrastructure for several scenarios
    - Supporting smaller often less sophisticated organizations
    - Supporting dynamic small Grids like personal Grids
    - Supporting dynamic security model to support for example the students in a training session with teacher having decentralized authority
    - Control sensor nets, mobile phones etc
    - Jini seems successful in several projects
- Need Peer-to-peer Grid infrastructure (implemented as JXTA with OGSI mediation for instance)
    - Capital Radio
- Need to customize Grid software to support special classes of Grids such as Campus Grids
    - One would expect in general that Campus Grids and e-Business would prefer commercially supported software
- Need to support in hardware and Grid software infrastructure fields with both large simulations and pre and post processing of data as in Environment and Earth Science areas
    - Queen Bee architecture (as in figure 6.4) with the e-Science Grid consisting of a swarm of highly distributed cost effective nodes together with central massively parallel machines for large simulations
    - Data assimilation will grow in importance and be naturally supported in such integrated data/simulation Grids
    - Database resources such as those at EBI need "computing-on-demand" attached to them for processing both "system" and "user" algorithms. There are issues of how resources charged; where they are placed (to avoid large data transfers), and how one certifies user programs for safe execution. Particle physics has a variant of this problem in a more structured resource-rich environment.
- Develop an architecture and infrastructure to integrate/federate multiple interoperating Grid Islands
    - Learn from database federation and migration of existing Jini/JXTA and other non Globus Grids to Globus
    - Integrate the different types of Grids desired and cope with natural organizational and application domains
    - Can expedite implementation of lightweight and P2P Grids
    - Integrates well with existing VPN (commercial virtual organization).
- Support for mobile code as well as mobile data
- Will federation really work for resources like those at CLRC shared between (10) different communities

8.1.3 Jini

8.1.3.1 Grid Prototypes Based on Jini

Lightweight Grid systems can be built using Java as the primary programming language and Jini as the underlying service-oriented infrastructure [Jini], as has been demonstrated in the ICENI [ICENI] and DiscoveryNet [DiscoveryNet-A] projects from Imperial College (see Sections 8.2.3.2, A.3.3, and A.7.2) and the JISGA project from Cardiff University [JISGA] (see Sections 8.4.3.3 and A.2.4.1.3). Jini's main strengths are that it is stable, commercially-supported, well-documented software with a large user base. In addition, Jini is freely-available and easy to install – Jini is a Java technology that can be installed on almost any modern computer. Although Jini was originally designed to support the integration of networked devices, it has been found to be useful as a general-purpose distributed environment. Jini not only provides easy integration with the Java language, but it also provides several desirable features for building a grid middleware. Fundamental to Jini, through its use of Java's Remote Method Invocation (RMI), is the declaration of a service interface. This interface is registered into Jini's Lookup Server with a

'lease' declaring the duration of the service's availability. Clients search the look up server for service instances based on the service type. The use of leases within Jini is recognition of the transient nature of services within a distributed (grid) environment. The event notification architecture within Jini allows any service failure to be trapped and handled. However, to use Jini in ICENI it was necessary to incorporate methods to hold and search additional service meta-data than that provided in the standard Jini model.

Use of Jini for Grids is discussed in the appendix, section A.2.1.1.

8.1.3.2 Jini in Agent-Based Grids

Jini technology from Sun Microsystems provides a distributed software environment that facilitates the dynamic inter-operation of devices, services and applications on a network [Jini]. The DARPA Control of Agent-Based Systems programme [CoABS-A] used Jini as the basis for the development of the CoABS Grid, which had the aim of demonstrating the run-time integration of heterogeneous agent, object and legacy software components into military applications. The CoABS Grid makes use of two important components of Jini:

- *look-up services*, which are used to register and discover agents and other services; multiple look-up services can be run for robustness and scalability;
- *entries*, which are placed in look-up services by agents to advertise their capabilities.

The CoABS Grid uses the underlying Jini infrastructure to provide a message-passing service for agents; a logging service to log both message traffic and other information; a security service to provide authentication, encryption and secure communication; and event notification when agents register, de-register, or change their advertised attributes. In demonstrations of the CoABS Coalition Agents Experiment [CoaxGrid] up to fifteen laptops were connected via the CoABS Grid with the system incorporating in the region of seventy heterogeneous agent systems and agent-wrapped legacy military systems.

8.1.4 JXTA

Use of JXTA for Grids is discussed in the appendix, section A.2.1.2.

8.1.4.1 Critique of JXTA for Building Peer-to-Peer Grids

JXTA is a communications middleware for peer-to-peer application development and is a set of open, generalised, peer-to-peer (P2P) protocols that allows any connected device to communicate, collaborate and share resources on an ad-hoc, pervasive, peer-to-peer virtual network [JXTA].

In posing the question as to whether JXTA is an appropriate infrastructure for building Grids, we are effectively asking two related questions. Firstly, is P2P an appropriate paradigm for Grid construction, and secondly, even if this is the case, whether JXTA is the most appropriate P2P infrastructure for this task. In answer to the first question, all popular Grids are currently based on P2P technology. Examples of this include compute Grids such as those using Entropia technology [Entropia] and data Grids such as Kazaa [Kazaa], Grokster [Grokster] and Morpheus [Morpheus] which were all initially built on Fasttrack's P2P protocol suite [Fasttrack]. The important point to realise here is that Grid computing in these forms has already entered the mass market to basic web and Internet users. These systems are simple to set up and new Grids are emerging daily. The existence of both compute and data grids using P2P technology gives weight to the argument that P2P methods can be used effectively to build certain Grids. The second question is whether JXTA is an appropriate platform for P2P Grid construction. Supporting JXTA is the fact that:

1. JXTA is the only standard attempt to formalise P2P computing.
2. There are numerous reference implementations for JXTA, with work in developing reference implementations for most popular languages. However, it should be noted that non-Java implementations generally have limited functionality, e.g. the JXTA C version currently only implements edge-peer functionality and cannot act as relays or rendezvous nodes.

3. JXTA has many developers (going by the number of downloads) and there are many products being developed using this platform.

The main counter-arguments to using JXTA as a basis for building P2P grids focus on the maturity of the software and scalability issues. JXTA is still in its early stages of development, and as such there are problems that can be encountered when using the technology. Connectivity is a main issue, every time a connection is made to the JXTA virtual network a peer typically sees a different topology, and more importantly, perhaps only a subset of the entire network (network partitioning). This means that some peer, known to some connecting peer, may or may not be accessible even if both exist upon the same network. Discovery of other peers and services can take time, and can also generate a lot of network traffic; this is due to discovery messages being propagated throughout the network. JXTA maintains a cache upon each peer, this holds addressing information about other peers and any other critical information, the way JXTA handles this cache appears to be inefficient, and we have experienced extreme amounts of disk thrashing as the cache is periodically maintained/updated.

Giving these noted problems, JXTA does still remain a viable option for building Grids if such problems with the implementation are resolved. JXTA also address several issues not considered currently within the Grid community. One significant issue is the nature of a JXTA peer. A JXTA peer can be a client, a server, a relay or a lookup server (i.e. a rendezvous node). In Web Services, the lookup server is located on a third party machine, e.g. using UDDI or a similar mechanism [UDDI-A]. Furthermore, JXTA peers can be dynamically organized as the network grows in a number of ways and employ the advantages of small world networks, where a large number of nodes can be connected in such a way as to minimize the number of hops a packet takes in order to traverse the network.

Organization of JXTA peers is independent to the underlying physical devices and connectivity and arranged within a *virtual network overlay*. Peers are not required to have direct point-to-point network connections and can spontaneously discover each other on the network to form transient or persistent relationships called peer groups that define specific behaviour for its peers. Current web services and OGSA specifications do not address these issues to a comprehensive degree. Much can be learned from the P2P research in this respect. OGSA, for example, is presently more concerned with the secure creation or deployment of services and their organization with respect to their transient nature. Another key feature of JXTA is the ability to traverse Natural Address Translation (NAT) systems and firewalls. In contrast, JXTA does not address how a service is created, it simply assumes such services already exist (as they do with file-sharing software for example). The dynamic creation of JXTA services would not be possible without the aid of an external toolkit, such as Globus.

JXTA operates at a higher level of abstraction than the current OGSA proposals and, in this respect, could have some advantages for use as a basis for grid construction. The basic application assumptions reflected in the Globus middleware where an application and/or its data are downloaded to remote computers for execution can easily be supported in JXTA for Java applications. However, for applications in other languages some kind of wrapping may be necessary.

It is probably inevitable that many Grids based on JXTA will appear over time. JXTA is however, not all things to all people. Many of the semantics of Globus Grid computing are absent from JXTA, or quite different in JXTA. This does not invalidate either approach. Users will ultimately dictate what services are required from Grids – many users are already experimenting with P2P in the form of JXTA and are finding success. This is certainly not a technology that can be ignored.

8.1.4.2 Use of JXTA in a Grid for Capital Radio Group

Capital Radio is the UK's leading commercial radio group, with greater revenues and profits than any other commercial radio company. This is achieved through a total of 20 analogue radio licences broadcasting to over half of the UK's adult population. Capital Radio Group is considering the use of JXTA to support a Grid-like infrastructure for data transfers between the group's 15 radio stations, which employ 50 servers and about 1000 clients. JXTA is deemed to be appropriate for this sort of data grid in which information

flow is multidirectional and, in many cases, time critical. A central server solution is inappropriate due to quality of service requirements because:

1. The stations need to be able to operate in isolation even in the event of network or server failures, so the grid must be robust, resilient, and reliable (RRR) as in an autonomic grid.
2. Many data transfers are time critical – a requirement that could not be met by a centralised architecture in peak conditions.

An important mode of use of such a grid is to be able to "compose" and distribute programmes using material produced at different radio stations and stored at different locations. Another important feature is to be able to be able to integrate local and regional advertising into programmes in a location-dependent way. There is also a need to be able to replicate and break up files (chunking) for better performance, and to provide a mechanism for persistent store administration. The JXTA Content Management System (CMS) [JXTACMS] is being considered for sharing files for these types of data grid problems.

8.1.4.3 JXTA GAT Binding for the Triana Project

Application developers are faced with a choice of a number of APIs and middleware systems that can be used to distribute their applications onto the Grid. This often leads to confusion and in some cases portability problems if a particular underlying technology does not succeed to its projected standard. To this end, an important advancement has been achieved by defining the Grid Application Toolkit (GAT) API [GAT]. The GAT provides an application-driven API and implements key bindings to the various underlying mechanisms for the implementation of this functionality. The GAT can also be dynamically switched at run time to utilize the functionality that exists on a particular platform or environment. Current GAT implementations include Web Services (OGSA to follow shortly), JXTA and local services for prototyping applications.

The Triana team [Triana-A] has implemented the JXTA GAT binding and is currently using this to prototype the distributed behaviour of Triana onto the Grid. Such a prototype, without any modification, will later be used to deploy Triana OGSA services using the GAT OGSA binding, when it is available for production use, just the same way as JXTA services are deployed now. Briefly, the Triana distribution mechanism is based on Triana services. Triana services can be implemented and deployed in a number of ways, e.g. as JXTA services, Web services, OGSA services, Jini services etc. Connecting to such services is specific to the underlying GAT binding being used. For example, JXTA pipes are created to communicate with JXTA services, while a Jini proxy interface to a remote RMI Triana service would be used for Jini.

8.1.5 .NET

Use of .NET for Grids is discussed in the appendix, section A.2.1.3.

8.1.5.1 Overview of .NET

Microsoft .NET is an XML Web services platform that provides comprehensive support for major open-standard Web services technologies. Various tools are available for reducing the complexity of implementing a Web service, and for providing support for messaging between such services. Visual Studio .NET provides a development environment for writing such services in C# and J# (Microsoft's version of Java), for instance. The importance of such tools is essential to enable a wider adoption and usage of Web services-oriented Grid software.

.NET enables users to create new Web services, or transform existing programs to Web services in a relatively simple manner and provides comprehensive cross-language support. Grid services that provide resources such as compute, data and application, especially those available from Windows platform, can be constructed based on the .NET platform and be accessed transparently. The latest enhancement package to .NET also provides the much-desired Grid features such as security management and improved performance on data transmission. In addition, together with Microsoft Internet Information Server (IIS) and the Active Directory technology, .NET can be used to create the hosting environment for transient Web services proposed in OGSA.

The .NET platforms also support a number of additional features that could be useful to construct Grid systems, particularly for enabling a number of different platforms to interact. The "Common Language Runtime" (CLR) in .NET provides an execution environment that can be ported to a number of different machine architectures, and supports the execution of programs written in a variety of different languages. The CLR adopts a similar approach to the Java Virtual Machine, for enabling code developed on one machine to be ported (without re-compilation) to another. The .NET platform also provides a Just-In-Time (JIT) compiler that allows dynamic performance improvements during the execution of a program, and is facilitated in this by the CLR. Microsoft has also made a version of the CLR open-source, to enable community input into this activity.

In August 2000, Microsoft Corporation, Hewlett-Packard and Intel Corporation co-sponsored the submission of specifications for the Common Language Infrastructure (CLI) and C# programming language to the international standardization organization ECMA- this work is active and ongoing [ECMA.NET]. Furthermore, the Mono project [Mono.NET] is bringing the .NET platform to Linux. Early demonstrators of this in action (at University of Southampton) include construction and deployment of peer-to-peer systems build on Windows and deployed on Linux. .NET is thus able to provide both cross-language and cross-operating system support.

8.1.5.2 Grid Middleware Based on .NET

It has proved straightforward to construct and integrate together Windows based web-services constructed using .NET and those from a variety of Linux web-service construction tools. This is occasionally necessary where codes only run on Windows, and is desirable where there is a requirement to improve resource utilisation on idle (Windows-based) desk-top PCs. Particular examples from the Geodise project [Geodise] of use of .NET include providing a .NET interface to Condor [Condor] and wrapping a number of database services using .NET. The .NET Web service enabled interface to the Condor system has been constructed in order to achieve programmatic, platform and language neutral access to the resources managed by Condor. It provides mappings between the computation toolkit functions and the resource management mechanisms of Condor. The interface is primarily an interpreter between XML messages representing the compute operations and the ClassAd language of Condor [Condor-ClassAds]. It also hides from the users details of resource management that are proprietary to Condor, so as to achieve the desired transparency. In addition to interactions with Condor, the service also implements general Grid service functions that are not provided by the Condor system, such as management of security and process status.

8.1.6 Grids and Virtual Private Networks

The UK e-Science Grid is one of the first Grids to attempt to deploy Globus outside of a private network. A number of security issues have emerged in the design of GT2 when deployed over large public networks such as the internet [Globus-B]. Essentially the problems reside in two areas:

i) the use of port mapping - whereby a well known port is used to negotiate a transient high level port used for the rest of the client/server session
ii) a requirement for a client to have a fixed IP Address for notification services

In practice, many institutions make use of technologies such as port blocking routers, firewalls, NAT (Natural Address Translation) and DHCP (Dynamic Host Configuration Protocol) to improve the security and manageability of their networks. GT2 is not compatible with these technologies because IP addresses are allocated dynamically and routed through a "proxy" public IP addresses, or the particular ports used by GT2 are blocked.

One solution would be to use VPNs (Virtual Private Networks) to create the private network environment that Globus is designed for. VPN's use tunnelling technologies so that clients behave as if on a single private network, even though traffic may pass over public internets. There are currently two VPN technologies - PPTP and L2TP which tunnel the network traffic in different ways. L2TP is the more secure and it can use IPSec to encrypt the data, and hence can make use of certificate based PKI. VPNs can also provide a solution to the fixed IP Address issue. Although the client machine may have a dynamically assumed or NAT IP Address, the VPN server could allocate a static (virtual) IP Address. If the VPN is based on L2TP/IPSec then this could be allocated by using a Grid certificate to authorise and authenticate the end user.

There are however a number of issues with VPNs:

i) firewalls may block the ports or protocols required by PPTP or L2TP. However, the required policy for the firewall management of participating institutions is far simpler (as only the ports and protocols for the VPN will be required to be open)
ii) vendor interoperability is still an issue, for instance in Oxford they have not managed to use the Microsoft VPN client to attach to a Cisco VPN server. This has potential impact if the Grid is to support heterogeneous platforms.
iii) availability of VPN clients on some platforms may be problematic - especially if the Grid is to encompass mobile devices
iv) Access to local resources - by default a VPN client will route all requests to IP Addresses outside the local subnet over the VPN. This may cause problems if the user wishes to access resources on a local firewall protected WAN (Wide Area Network). If the resource is outside the immediate subnet but within the WAN firewall the VPN client will attempt to route the request over the VPN and outside the firewall, hence the request maybe blocked by the WAN firewall. It is theoretically possible to manually configure the client routing tables to cope with this, but in practice this depends on the VPN client/platform and can be difficult for a novice to setup correctly.
v) Possible problems if the user is required to use multiple VPNs. It is possible that is the user needs to access multiple Grids simultaneously or is a member of multiple VOs that they may need multiple access to multiple VPNs simultaneously. VPNs are becoming common outside of Grid applications (for example as a mechanism for protecting wireless networks) and it might be that a user needs to simultaneously access a Grid VPN as well as non-Grid VPNs. In theory, it is possible to establish simultaneous VPN connections, however this can require expert manipulation of the routing tables to get working correctly. Problems can arise if different vendor VPN clients are installed on the same machine, for example it has been observed that Microsoft VPN connections no longer work when the Cisco VPN client is installed (but work once the Cisco client has been uninstalled).

The original version of this material can be found in the appendix, section A.2.1.4.2.

8.1.7 Grid Federation

Grids need interoperability standards to support the linkage of distributed resources developed by an inevitably distributed team. These standards for core Grid software come from IETF, W3C, OASIS, and especially the Global Grid Forum. In each application area, there will be other domain specific activities. There are inevitable conflicts between the essential standardization efforts and the continuing evolution of the Grid and the radically different requirements for different Grids described in section 8.1. So although continued support of OGSA and OGSI is essential, exactly how this should be applied is less clear. This is an important issue for the e-Science program which currently has several rather different Grid infrastructures

1) Globus Toolkit 2 based Grids
2) Largely Web Service based environments without significant use of existing GT2 capabilities. Many of these projects will use OGSA-DAI.
3) Grids based internally on non Globus or Web Service distributed object technologies such as Jini

All categories could be transitioned to GT3 but there are different strategies that could be considered. First one could fully adopt GT3; as a second option (attractive perhaps in the Web service based Grids), one could adopt the GT3 OGSI implementation and their hosting environment but not their full range of OGSA services. Finally one could use OGSA as a federation standard at the boundaries between disparate Grids. Figure 8.1.1 illustrates the difference between the layered and island approach to implementing Grids. Each relies on standards like OGSA. In the layered case of fig. 8.1(a), all services explicitly or implicitly implement the given standard; in the “island” or federated scenario of fig. 8.1 (b), each constituent implements “native” versions of services while at the boundary between the islands, a mediation service – using OGSA perhaps as a lingua franca – the different service implementations are reconciled. The need for the “island” model was discussed in section 8.1.1. Two examples are compelling; the first corresponds to the need to include existing legacy systems in a Grid where it is impractical to convert to OGSA on a

service by service basis; this is likely to be important in business and government areas. The second example is more relevant for e-Science and corresponds to the interest in lightweight Grids which are easy to deploy and possibly offer features developed by the peer-to-peer community. Considering a Grid built around the open source Sun Microsystems technology JXTA [JXTA], this is built on message-based services but with a rather different set of core mechanisms from that envisaged in OGSA

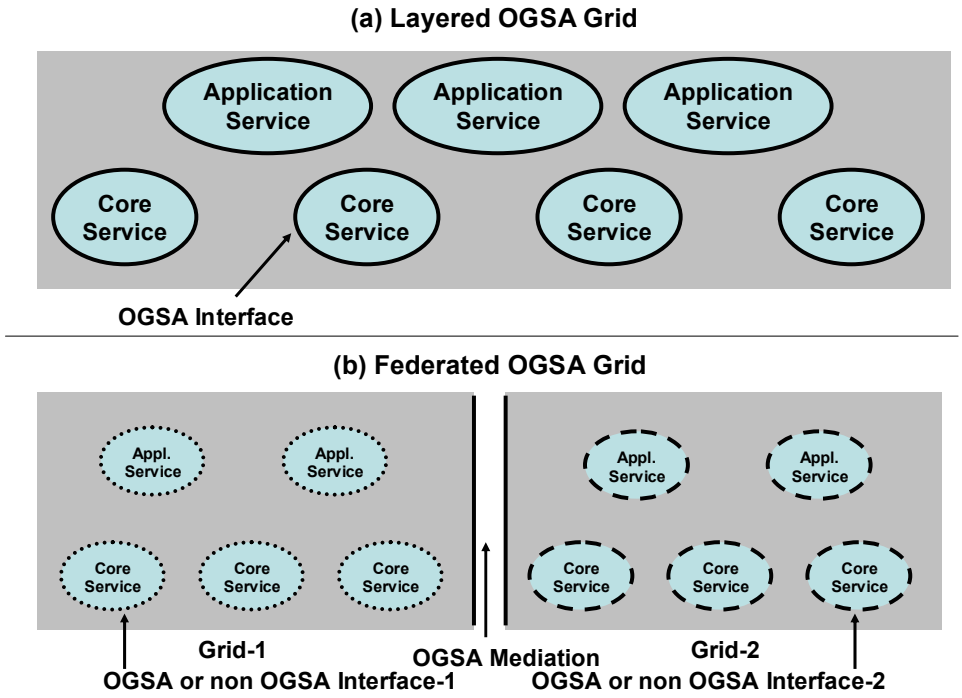


Fig. 8.1 (a) A classic layered Grid architecture with all services supporting a common interoperability layer; 8.1(b) a federated Grid made up of two individual Grid islands using idiosyncratic internal service protocols and interfaces. The Grids are integrated into a single OGSA Grid using a mediation layer at their boundaries

In this section, we also collect technologies which are connected to building federated Grids [Pierce03A]. The natural approach to this is based on inter-Grid gateways that intercept messages travelling between Grids and do any necessary conversions. This is similar to the route that CORBA went down with the introduction of the General Inter-ORB Protocol. Note the federation idea depends critically on the message-based service model underlying Web and Grid services. Further many runtime environments have identified message oriented middleware (MOM) as an appropriate architecture [MOM] and defined messaging as a critical layer in a middleware architecture. Such Grid messaging technology is discussed in the next subsection and the mediation discussion builds on this abstraction.

8.1.8 Dynamic Configuration (Autonomic Computing)

Dynamic re-configuration is the process of changing an executing system without stopping execution and with minimal re-computation after the system has been reconfigured. It can take two forms:

1. Replacing executing modules in the system with different modules. This is required for non-stop systems where the software must be upgraded without taking the system out of service.
2. Changing the platform nodes on which modules of the system are executing.

This latter form if dynamic re-configuration is the most relevant for the Grid, at least in the foreseeable future. There are two forms of this dynamic reconfiguration that are relevant for the grid.

1. Application restructuring to take advantage of additional execution platforms that may become available. For example, if a parallel application can run on any number of nodes, then the number

of nodes actually used may change during the execution depending on availability, cost, etc. This is required to realise the far-sighted vision of the Grid.

2. Load re-allocation where executing jobs are moved in execution from one node to another. This might be because a node has failed or shows signs of failure, because a cheaper resource has become available or as part of an overall load balancing strategy.

Load re-allocation is currently the task of system managers and usually involves stopping a job before moving it to another node. If we are to develop an autonomic computing system, we need to automate this process and allow the system itself to allocate and re-allocate jobs to nodes.

The problems of dynamic re-configuration are non-trivial and current grid middleware does not provide any support whatsoever for this process. The most promising technology to apply here is probably reflective middleware where the middleware can have awareness of its own actions. There is then the possibility of developing strategies for dynamic re-allocation.

It is unlikely that current techniques of dynamic re-configuration are applicable without change to the Grid; reflective middleware is still a research topic. Therefore, my view on this is that further research into dynamic re-configuration and the Grid is required before the middleware 'gap' can be filled.

The original version of this material can be found in the appendix, section A.2.1.4.1.

8.2 e-Science Runtime and Hosting Environment

8.2.1 Introduction

Any computing model involves multiple aspects. OGSA defines interfaces and protocols while in section 8.1, we discussed overall architectural principle and the mediation between different Grid islands. In section 8.4, we describe the application programming model in terms of workflow or the orchestration and integration of services. This assumes a "two-level model" of programming the Grid where the software defining a service is programmed using conventional (non Grid) technologies but with calls to some library that implements the Grid and Web services ports. This library reflects the runtime in which the services operate and this runtime will also impact how one implements particular workflow models.

The Grid runtime is roughly equivalent to what is called the hosting environment in some Grid discussions. If one has multiple federated Grids as discussed in Section 8.1, undoubtedly one can expect this runtime to vary from Grid to Grid but even within a single Grid this runtime can vary over the different devices used to form it. A Grid containing servers, desktops and PDA's would probably have different runtimes on each device to reflect their intrinsic capabilities and perhaps their native operating system (say Linux, Windows, and Palm). OGSA and OGSI are aimed at insulating the developer as much as possible from the different runtimes but still the issue is important.

Each of the important development environments (.NET [OGSI.netUVA], GT3 [Globus-C], Enterprise JavaBeans (EJB) [EJB], Java Servlets [Axis], JXTA [JXTA], Jini [Jini]) has different natural runtimes. In this section, we present interesting activities in this area which could lead to more productive or higher performance Grid software development. Interestingly whereas the middleware community has active research in this area, there has not been much direct discussion of the Grid or e-Science runtime by any community and in fact much of the standardization work has perhaps unnecessarily de-emphasized its importance. Certainly continued research in this area is warranted but there also seemed to be consensus in some areas that could usefully be early projects in any proposed e-Science middleware initiative. Highlights include the development of frameworks that hide the details of OGSI [OGSI] from developers. This could not only automate generation of OGSI features but also allow pure state free Web service implementations to co-exist with OGSI stateful services. Similarly one could use a Grid runtime framework to allow a given set of services to transparently use either a simple point to point model or a more sophisticated JMS (Java Message Service) [JMS] style notification implementation.

Enterprise JavaBeans [EJB] were developed to robustly support business applications and have excellent support for database transactions typical of this area. This model is not very suited for the demands of scientific computing that often involve not transactions but rather large bandwidth dataflow. Thus one could consider redesigning the EJB model to better support e-Science. In doing this one should be mindful

of the benefits of simplicity; the need to deploy EJB's and similar technology could be a major barrier to the use of Grids outside large organizations.

8.2.2 Summary of Gaps

- There were several discussions of the development and runtime environment of Grid components. This is an area that is not often explicitly discussed but is implicitly defined by a choice of a framework (.NET GT3 EJB Servlet). This is related to the hosting environment but note one can have different development/runtime environments for different services. In particular the hosting environment should be distinguished from Portals which are the user (not the service) development environment. Here we discuss the environment in which the services invoked by Portals are developed.
    - "Scientific Container" or the Science optimization of Enterprise JavaBeans (EJB)
    - Note remark that business environments like EJB optimized for short transactions where as science needs high bandwidth dataflow
- Need better invocation framework hiding differences between Grid and Web services (SessionID versus factory), messaging etc.
    - Note some consider that only state-free Web Services can scale and avoid factories and explicitly spawning stateful services for each service invocation. Rather this is managed on the server hiding session dependent processes from the rest of the Grid. Cookies and/or user login link a particular client (accessing service) to a particular set of data on Web service. The invocation framework hides e-Science from this uncertainty in the evolution of Grid services
- Need more language bindings other than Java for service infrastructure
    - Perl is very important in Bioinformatics
    - C C++ Java
- Need Schema checking in service hosting environment
- Ability to stage services; instantiate from factory on general resource
    - Current factories (in GT3) do not support remote instantiation of child services.
- New ideas from middleware community could be used in this arena to provide adaptable containers built from lightweight components
- Use wrappers supported by runtime to support provenance and security (at Southampton) – what other wrappers are useful?
- Need context sensitive middleware with policy driven binding
- Asynchronous messaging is needed for Autonomic Grids
- Add Messaging and Transport layer with links to both higher middleware and Network layers
    - Include QoS with generalized Network Weather Service
    - The virtual private grid VPG of [Pierce03A] (see VPN discussion in section 8.1.6) would be supported at this level
- One needs scalable fault-tolerant management framework including support of notification and provenance

8.2.3 Selected Snapshots of Current Activities

8.2.3.1 Invocation Framework from IT Innovation

Users of the grid will access its capabilities via client-side applications – grid browsers or more sophisticated problem solving environments. To develop such applications one must have an API for calling operations that are themselves provided by grid services. Much attention has been given to standardizing grid protocols, and more recently grid service functionality, but it is also crucial to provide a good invocation framework. The key is for the invocation framework to be capable of handling arbitrary grid services, just as web browsers can access arbitrary web "pages".

IT Innovation [ITInnovation] has worked on this problem over many years in a series of EU 4th Framework and 5th Framework distributed computing projects, mainly based on CORBA. Recently the results of this work have been distilled out to provide a web service invocation framework for myGrid [MyGrid-A],

which was further, refined and made secure for Comb-e-Chem [CombeChem]. The basic invocation interface generates and transmits SOAP messages based on:

- the URI of the WSDL [WSDL] document defining the service;
- the PortType of the WSDL containing the desired operation;
- the operation name of the WSDL and
- the operation argument values.

IT Innovation's framework is therefore similar to the Web Service Invocation Framework (WSIF) developed by the web service community [WSIF]. WSIF provides a wider range of service invocation methods based on SOAP, RMI or JMS protocols. The main differences are that IT Innovation's software:

- is more stable, being based on earlier (CORBA) dynamic invocation frameworks;
- handles UDDI lookups [UDDI-A] as well as straightforward web service invocation;
- supports digital signing and authentication of SOAP messages using methods defined in the W3C SOAP Digital Signature Extension Note;
- handles HTTP and HTTPS proxies transparently and correctly.

The software is available under LGPL license terms, and is being used by two EU grid projects (GRIA [GRIA] and GEMSS [GEMSS]) as well as the UK myGrid [myGrid-A] and Comb-e-Chem [CombeChem] projects. IT Innovation intends to develop the framework in GEMSS to add in-line support for W3C Web service WS-Security/WS-Policy extensions [W3C].

The original version of this material can be found in the appendix, section A.2.2.1.1.

8.2.3.2 ICENI From Imperial College

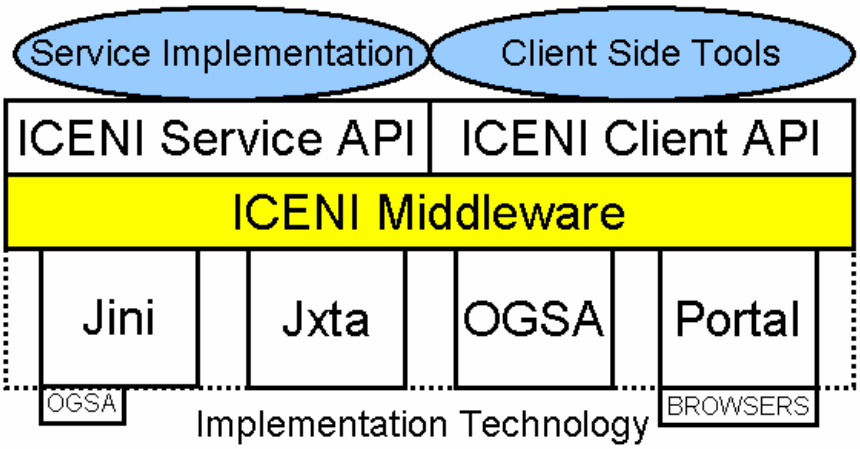


Fig. 8.2: Architecture of ICENI Grid Component Model

Following UCL's experiences with the initial ICENI prototype and the emerging needs from collaborators within the UK e-science programmes pilot projects recent effort within the LeSC Grid middleware group [LeSC] has concentrated on the redevelopment and refactoring of ICENI [ICENI]. Its architecture is given in fig. 8.2 and described below:

This approach allows the application developer to implement an ICENI service using an architecture neutral interface thereby providing a consistent interface between underlying service implementations. The meta-data annotating the service may be exposed to the underlying service architecture as appropriate. The service and discovery interfaces within ICENI may be realised through a number of service oriented

infrastructures such as Jini, OGSA or JXTA [Jini] [OGSA] [JXTA]. Alternatively, the services may be accessed through a portal infrastructure. In each case the same service implementation and meta-data is reused. We currently have an implementation of the ICENI service API working with Jini as the underlying service architecture where Jini services may be exposed to an OGSA (Open Grid Services Architecture) environment, and are prototyping pure JXTA and OGSA implementations.

Within ICENI we separate out the description of our architecture into two abstractions:

- a *resource* which has a capability for action and a defined behaviour and performance if invoked
- a *service* which exposes the resource abstraction through an initial service level agreement (SLA).

This information is all encapsulated within an extensible meta-data schema. As the Grid community moves towards service oriented architectures and services become pervasive, a user will potentially be exposed to a plethora of services. Within this context they will only wish to see the services that they are able to access. Therefore, from a user's perspective the virtual organisation becomes a view (or slice) of the service registry that they have access to. Likewise an administrator will have a different view of a service registry – the services that they are able to administer – than a user.

The ICENI middleware also provides an infrastructure to define and deploy an augmented component programming model being developed within LeSC to tackle issues relating to effective application deployment within the Grid. Software components are annotated with meta-data relating to their interface, their behaviour (both inter and intra component workflow) and performance. The ICENI scheduling framework is used to provide an optimal mapping of the components onto the grid resources by using this meta-data. Once deployed these components appear as services within the ICENI framework.

Part of the standard service interface within an ICENI services is the ability to create new service instances through a Factory paradigm. This is an action that may be invoked on a service and is therefore subject to the configuration using the previous access rules. We currently envisage two factory methods: the first creates a new service instance with a set of access rules specified by the user alone, while in the second service instance the user's access rules are verified before checking those of the service administrator. We thereby present a very flexible approach to building 'ad hoc' virtual organisations through the sub-contracting (or delegation) of service access to entities known to the user, but not necessarily the service owner. This approach allows very flexible, extensible and dynamic organisations to be built but reduces the control that an administrator has on the 'ultimate end user'. This is obviously a matter for deployment configuration.

The appendix in sections A.2.2.1.2 and A.3.3 has substantially more detail on ICENI.

8.2.3.3 Messaging Infrastructure

The NaradaBrokering infrastructure [NaradaBrokering] is designed to have some of the features needed by an asynchronous message-based execution environment. It virtualizes destinations and transport protocols; supports publish-subscribe methodology with XML Schema based topics; can link to Network performance modules and traverse (some) firewalls. Systems like IBM's MQSeries [MQSeries], Microsoft's Pastry [Pastry] and JXTA [JXTA] exhibit similar capabilities. The new Web Service messaging standards [WSAddressing] [WSReliableMessage] [OASISWSRM] will be important in driving this area.

Note this low-level publish-subscribe capability is different from that needed by the OGSA higher level notification schemes [OGSA] as discussed in section 7.5.

The original version of this material can be found in the appendix, section A.2.2.1.3.

8.2.3.4 Lightweight Middleware

Recent research in the middleware community has investigated lightweight and flexible approaches to the implementation and deployment of system software in general, and middleware in particular. This 'lightweight middleware' research has two main goals: *i*) to facilitate the construction, deployment and evolution of (middleware) platforms and services in their full generality, and *ii*) to facilitate the run-time

management, adaptation, and dynamic reconfiguration of such platforms and services. By 'platforms and services' we mean *both* low-level, fundamental, 'platforms' (or 'runtimes') such as the software underpinning Web-Service invocation, CORBA-style invocation, media-streaming, tuple-space platforms, etc., *and* higher-level 'services' such as security, persistence, meta-data lookup, workflow, etc. Further, this research addresses the construction, deployment, evolution, and management of both existing (standardised) platforms (e.g., OGSA [OGSA], EJB [EJB], JXTA [JXTA] etc.) and novel, niche platforms and services that might be useful for particular application areas or deployment environments (e.g. involving alternatives to the SOAP protocol [SOAP] when sending bulk, structured data over a wireless network or doing multi-peer communication). Essentially, the research is working on toolkit support for developing, deploying and managing middleware and system software of all kinds.

The approach most commonly taken in this work is to build systems in terms of a well-founded, low-level, runtime *component model*. Such component models are lightweight in nature and add very little in terms of performance overhead and memory footprint. As an example of such a model, we briefly describe Lancaster University's OpenCOM [Clarke01A]. OpenCOM is language independent (i.e. components written in various languages can be combined arbitrarily), and system independent (it runs on Windows and UNIX platforms and can even run on bare hardware—this latter is achieved by recursively implementing parts of the OpenCOM runtime itself in terms of OpenCOM components). OpenCOM supports three main concepts: components, reflection, and component frameworks. *Components* serve as basic building blocks for constructing systems by composition (as advocated for example by Szyperski [Szyperski98A]). *Reflection* then provides means to discover the structure and behaviour of component compositions and to adapt and extend these at run-time. Finally, *component frameworks* have the role of imposing domain-specific structure on component compositions, and ensuring architectural integrity during periods of adaptation/ reconfiguration. Examples of such 'domains' (taken from the Lancaster work) could be a 'pluggable' protocol stack framework, a framework implementation of the CORBA Portable Object Adapter, a framework implementation of application level multicast, or a framework for local resource management of threads, memory, and communication endpoints.

Typically, these component models only comprehend the scope of a single address space. The idea is that where inter-address space communication is required, *the required middleware to achieve this is itself built in terms of components*. This enables great flexibility in deployment For example, a component framework implementing the IIOP protocol can be dynamically deployed into an address space to enable a set of 'application' components to interact. Subsequently, a set of components that implement a SOAP engine can be incrementally deployed if communication with a web service becomes necessary. Note that in this approach, the boundary between 'operating system', 'middleware' and 'application' is blurred: all of these are simply sets of components that are combined and deployed as required to meet an application need.

It is important to emphasise that existing commercial component models such as EJB and the CORBA Component Model are *not* what we are talking about. These are used to build *applications only*; they are too heavyweight and prescriptive to implement systems themselves. The same applies to the Grid-oriented component models that have so far been developed, e.g. ICENI from Imperial College [ICENI].

The main forum for this 'lightweight middleware' research is the ACM/IFIP/USENIX 'Middleware' series of conferences. Apart from the OpenCOM work mentioned above, other examples of research efforts following a similar approach are as follows: THINK [Fassino02A] and Knit [Reid00A] are component models that have been used to build operating systems; Knit has additionally been applied in programmable networking environments, as has Click [Kohler99A]; K-Components [Dowling02A] [Dowling03A] is a component model that has been used primarily for real-time middleware; LegORB and Universal Inter-Connect (UIC) [Roman00A] are used to build middleware for ubiquitous computing environments; JavaPod [Bruneton00A] is a Java-specific solution for building middleware.

Although it has been applied in a wide range of application environments, the 'lightweight middleware' approach has not yet been applied in anger in Grid environments. Nevertheless, we believe that the approach has a lot to offer the Grid. It provides a well-founded basis for the flexible decomposition, recomposition, and dynamic reconfiguration of Grid middleware that is arguably much better suited to middleware deployment in large scale, heterogeneous, and dynamic environments than are today's

monolithic and 'black box' solutions. For example it has the potential to facilitate the provision of QoS-aware, adaptive, and self-managing platforms, and to help provide structure and architecture in large scale systems. However, considerable research is required to reap the potential benefits of the approach in the Grid environment.

The original version of this material can be found in the appendix, section A.2.2.1.4.

8.3 Security Infrastructure

8.3.1 Introduction

This was not a focus of our work because there is a separate task force in this area [UKeSSTF]. However we got several comments in this area. A particularly interesting concept is that of a federated security architecture which is developed in a white paper by Chivers [Chivers03A] and these ideas are placed in a broader security context in the report of the Security task force. A summary of the Chivers report can be found in the appendix, section A.2.3.1. In the security area, we note the commonalities between Grid islands and VPN's (Virtual Private Networks in section 8.1.6) and suggests that integrating both technologies and processes (such as the Security model) could be useful. This concept is explored more fully in a white paper developed by Fox and Pierce from Indiana [Pierce03A].

8.3.2 Summary of Gaps

- Some features like certificate infrastructure are related to particular Globus technologies but reflect the field which has no consensus good broad solutions. Areas of importance include certificates, authorization and policy
- Need semantic firewalls to mediate trust/security between administrative domains
- Authorization tools (VOMS [EDGWP2VOMS], CAS [GlobusCAS]) and Policy (Permis [Permis], Akenti [Akenti]) have not led to consensus adequate solutions
- Some consider security has too much attention and others that identification of "patterns" or "scenarios" of importance is a fruitful approach. Perhaps there are important patterns in e-Science (such as setting up a virtual classroom) for which issues can be addressed allowing both better security and more user convenience.
    - We need single sign-on as critical concept and better support of roles
    - VPN is an interesting pattern
- A VPG (Virtual Private Grid) [Pierce03A] could help industry work with Grids as VPNs are accepted way in Industry of extending the Enterprise infrastructure.
- Current certificate infrastructure will not scale to any where near needed number of users
- Unicore's [Erwin02A] security infrastructure does not use proxies and has good tunneling on port 443
- WS-Security promising but roadmap unclear
- Public domain firewall tunnelling and VPN software could be important. This is discussed as part of a proposed VPG [Pierce03A]
- Support delegation
- Many of the current difficulties with security can be addressed by federated security architecture [Chivers03A].

8.3.3 Current Globus and related (EDG GriPhyn Condor) Technology

- Globus treatment of firewalls requiring changes in security policy (opening specific ports and resources) is insufficient
- Globus treatment of authorization insufficient
- Current certificate infrastructure too clumsy for broad deployment
- Globus CAS needs distributed servers in a single VO – current single server model does not function acceptably [GlobusCAS].

8.3.4 Possible Future Projects

8.3.4.1 Harden VOMS from EDG

EU DataGrid has developed and put into production tools for managing Virtual Organisations (VO) via a convenient web interface, for allowing users to digitally sign Acceptable Use Policies (AUP) via the web, for publishing VO membership and for automatically constructing local access control policy files. This system is used with a Pool Accounts extension to Globus to allow users to "join the Testbed" without any manual intervention by site administrators - for example, to verify AUP acceptance or to create new accounts. The system also includes a Virtual Organisation Membership Service (VOMS) [EDGWP2VOMS], which permits users to obtain signed attributes proving VO membership and roles, and Grid Access Control Lists (GACL) and a toolkit to manipulate them, which are used to specify policies controlling access to resources in terms of certificate identities and VO memberships.

All of the authorisation tools are potentially portable to other Unix variants, and would quickly benefit from being packaged for these platforms. Some development work in generalising the web interfaces to the VO management and AUP web systems would allow them to be applied to non-Particle Physics projects. Developing more detailed end-user documentation, including overviews of the security implications of different uses of the system, would make the VO and Pool Accounts approach more attractive to sites which have traditionally relied on static accounts and non-digitally signed AUP documents. GACL policy files and toolkit are designed to interoperate with other Grid authorisation systems, and this is an active area of co-ordination through the GGF Authorization Working Group [GGFAuth] (co-chaired by one of the GridPP [GridPP] developers) Consequently, this work would be ideally placed to influence this standards process, and produce software extensions to the EDG tools, to add support for other authorisation systems favoured by other UK Grid projects.

The original version of this material can be found in the appendix, section A.3.2.1. This is related to virtual organization management discussed in the appendix, section A.2.9.1.7.

8.4 Workflow

8.4.1 Introduction

Workflow is an established methodology for business process management, and in this context has been defined by the WfMC (Workflow Management Coalition [WfMC]) as follows:

"The automation of a business process, in whole or in part, during which information or tasks are passed from one participant to another for action, according to a set of procedural rules."

This definition can be conveniently adapted for e-Science by replacing the word "business" with "scientific". With reference to this definition of scientific workflow, the "participants" are usually compute- or data-oriented services, and the information and tasks that are passed from one to another define the data flow and control flow of the workflow. In general, the "participants" in a workflow can be arbitrarily geographically distributed, and the data and control flows span organisational boundaries; hence, workflow is well-suited for describing both e-Science and e-Business applications and activities.

Workflow lends itself readily to a graph-based representation in which the nodes of the graph represent activities or services, and the directed edges of the graph represent either data flow or control flow. Another common way of representing workflow is with an XML document that conforms to the schema of some workflow definition language.

An attractive feature of workflow is that it is inherently hierarchical in the sense that a workflow consisting of several nodes can itself be represented by a single node that can then be used in other workflows.

Two important aspects in the use of workflow in e-Science are workflow composition and workflow enactment. These will now be considered in more depth.

8.4.1.1 Workflow representation and composition

The standard way of representing a workflow is with an XML-based workflow definition language although there is, as yet, no agreed standard for this. For example, in e-Business BPEL4WS (Business Process Execution Language for Web Services pronounced "bepple" [BPEL4WS]) has been proposed by IBM and Microsoft [BPEL4WS], and XPDL (XML Process Definition Language [XPDL]) by the WfMC [WfMC]. In e-Science a number of workflow description languages have been developed, including SWFL (Service Workflow Language [SWFL]) at Cardiff, and GSFL (Grid Service Flow Language [Krishnan03A]) at Argonne National Laboratory. Whether a workflow description language designed for e-Business can be used effectively for e-Science is an open question. However, any workflow description language for e-Science must be able to express the programming abstractions, such as conditional and loop constructs, commonly used in scientific applications.

Whatever XML-based workflow description language is used, constructing a workflow with a text editor is a tedious and error-prone task. However, since a workflow can be represented as a directed graph, visual programming techniques can be used to construct workflows. The services comprising a workflow can be linked within a visual service composition environment (VSCE). Such an environment typically provides a mechanism for service discovery through which a virtual service repository is populated. An interface allowing services to be dragged from the repository and dropped onto a canvas, where they can be connected with other services through data and control links, is also a common feature of VSCEs.

The connection of services through data and control links within a VSCE is often referred to as providing "plug-and-play" capabilities in the sense that the environment should allow only services with compatible interfaces to be connected. In general, this desirable feature gives rise to some difficult challenges since not only must service interfaces be syntactically compatible, but also they must be semantically compatible. Syntactic compatibility requires the data types of data items flowing into a target service to be the same as the data types of the data output by the source services. Common data types will be defined in some XML namespace that everyone is assumed to make use of, and as long as these are the only data types used there are unlikely to be any semantic difficulties (except issues such as ensuring that units are compatible for non-dimensionless quantities and, where appropriate, converted). One form of semantic compatibility arises when more complex data types are defined in different XML namespaces. Some mechanism is then required to determine if an output from one service is semantically compatible with a particular input of another service, i.e., if the two data items have the same meaning (or sufficiently similar meanings) even though they have different names. This is one aspect of the more general problem of determining and comparing the behaviours and competencies of interacting services. The use of ontologies and agent-based mediation are two common approaches used to assess semantic compatibility.

A VSCE needs access to service descriptions given, for example, in WSDL (Web Service Description Language [WSDL]) to determine the syntax of a service interface. Additional metadata in needed in the service description to describe the service semantics and provenance.

Another factor that complicates the "plug-and-play" model of service composition is that the services in a workflow may not be bound to specific service implementations at the time of composition. This is the case when workflows are constructed based on the semantics of services, without reference to their interfaces. In fact, the binding of services may only occur dynamically at runtime. Thus, it may be impossible to check if two services are compatible when they are composed together. One solution to this is to "compile and link" workflows within the VSCE to check if they are feasible (i.e., the interacting services are compatible and discoverable). This doesn't guarantee the services will be discoverable in the future, but this opens up a whole new area in service lifecycle management.

8.4.1.2 Workflow enactment (Compilation and Execution)

After a workflow has been created in a VSCE and converted into an XML workflow description document, this document may then be submitted to a workflow engine for execution. The workflow engine needs to

convert the XML document into an executable form (a compilation), and discover and schedule services. These tasks lie at the heart of any service-oriented architecture for Grid computing. The scheduling may involve computational economy concepts, and load balancing, quality of service, and performance considerations. Where parallelism exists in the workflow the scheduler should be able to exploit it. This issue is closely related to the Grid runtime environment discussed in section 8.2.

8.4.2 Summary of Gaps

- Need a flow based scientific workflow engine and clarify differences between e-business and e-Science needs for workflow
- Workflow both in implementation and design is rudimentary; need both workflow languages (like BPEL4WS) and workflow enactment engines
- This is affected by runtime used i.e. Enterprise JavaBeans and Scientific JavaBeans might support different workflow models
- Integrate ontology support in workflow engines
- Need parallel execution and dynamic discovery

8.4.3 Selected Snapshots of Current Activities

8.4.3.1 Workflow at Southampton

Under development by IT Innovation [ITInnovation] as part of the myGrid project [myGrid-A], the *Workflow enactment engine* is a standalone component able to execute workflows expressed in a syntax close to the IBM WSFL language [WSFL].

- It will support dynamic discovery and invocation of services;
- It will support parallel execution of multiple threads;
- It will be deployable as a Web Service.

The original version of this material can be found in the appendix, section A.2.4.1.1.

8.4.3.2 Workflow at Newcastle

The ability to capture and enact computations is important for all the e-Science projects. Existing development of a workflow execution engine in the myGrid project (by IT Innovation [ITInnovation]) is the basis for the Workflow Enactment Grid Service (WEGS) for the Core Grid Middleware. The current myGrid enactment engine is based on Web Services, and so NEReSC [NEReSC] is porting it to be a Grid Service.

The WEGS in the initial release of the Core Grid Middleware will accept workflows written in WSFL (Web Services Flow Language [WSFL]). However, NEReSC is already evaluating the possibility of providing support for BPEL4WS (Business Process Execution Language for Web Services [BPEL4WS]) in addition to WSFL.

NEReSC is planning to follow the work of the GGF working groups in the important area of workflow/business process composition and intends to provide support for the Grid workflow standard when that is available.

The original version of this material can be found in the appendix, section A.2.4.1.2.

8.4.3.3 Workflow at Cardiff

Research at Cardiff University in the workflow area was focused mainly on the development of an XML-based description language for scientific service-based workflows, and on an execution environment that takes a workflow as its input, and supervises its execution. The XML-based description language is called SWFL (Service Workflow Language [SWFL]), and it extends WSFL (Web Service Flow Language [WSFL]) by supporting programming constructs, such as loops and conditional execution, commonly used

in scientific programming – these are the same loop and conditional constructs found in the Java language. A loop in SWFL can be tagged as sequential or parallel to indicate whether different loop iterations can be run in parallel. In addition, SWFL permits very general data link mappings that allows a greater degree of control over the handling of complex data objects that are input to a service, compared with that available with WSFL. The draft XML Schema for SWFL is given in the appendix A.4.1. The execution environment is called JIGSA (Jini-based Service-oriented Grid Architecture [JISGA]), and as the name indicates, is based on Jini for its implementation [Jini]. An SWFL document is submitted to JISGA as its input. Using the WSDL documents of the services referenced in the SWFL, JISGA then creates a Java "harness" code that invokes the services, mediates their interaction, and returns results to the submission environment. JISGA compiles the harness code and runs it, thereby executing the workflow in the original SWFL document. A JISGA system is made up of an arbitrary number of *WorkflowEngine* and *JobProcessor* services. A *WorkflowEngine* service receives the SWFL documents submitted to JISGA and decides how they should be handled. A workflow submitted by a client can either be processed in blocking or non-blocking mode, and can either be processed sequentially or in parallel, with details given in appendix, section A.2.4.1.3. The *WorkflowEngine* service directly manages the execution of blocking sequential jobs. The *JobProcessor* services are responsible for the processing and execution of all other types of workflow submission. The main tasks performed by a *JobProcessor* service are to break down workflows to be processed in parallel into sequential sub-workflows, and then to manage their execution. JISGA makes use of JavaSpaces as a shared memory mechanism for returning the results of non-blocking workflow submissions and for communicating between interacting services.

8.4.3.4 Workflow at IT Innovation

IT Innovation [ITInnovation] has met a requirement for workflow enactment in numerous projects going back to 1996. The original work was concerned with scheduling of task-graphs across distributed cluster systems in the EU projects PROMENVIR [PROMENVIR], HPC-VAO [HPC-VAO], TOOLSHED [TOOLSHED] and more recently DIAMANT [DIAMANT]. The task graphs could contain parallel or non-parallel branches, as in figure 8.3.

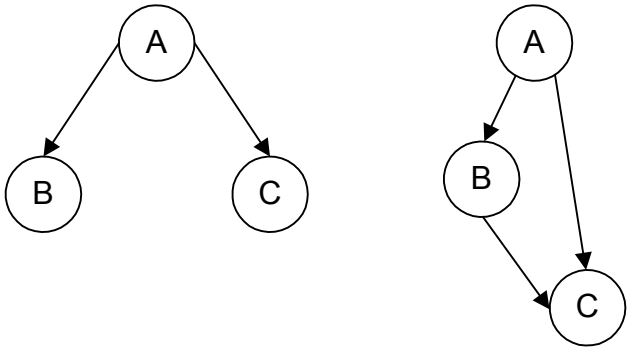


Fig. 8.3: Typical Task Graphs used in Workflow

In the UK e-Science programme, similar needs have arisen in orchestrating operation sequences on remote (i.e. distributed) grid services. IT Innovation's contribution to myGrid has focused on this issue, and created a new enactor that is also being used and developed in three European projects including the Grid projects GEMSS [GEMSS] and GRIA [GRIA]. The new enactor is structured as in figure 8.4.

The separate parsing stage allows a variety of workflow languages to be supported (myGrid is using WSFL [WSFL], but other projects use bespoke languages, and in future BPEL4WS will be supported [BPEL4WS]). The enactor core handles task dependencies and multi-threaded execution of parallel branches, and the invocation framework described in Section 8.3.3.3 allows tasks to be implemented as calls to arbitrary web or grid services.

The workflow enactor also allows "special" functions like UDDI searches [UDDI-A] [UDDI-B] and user selection from the results to be supported. These are used to support "late binding" of workflow tasks to

remote services. In future, we would like to enhance this aspect exploit new "semantic grid" paradigms for service discovery and binding.

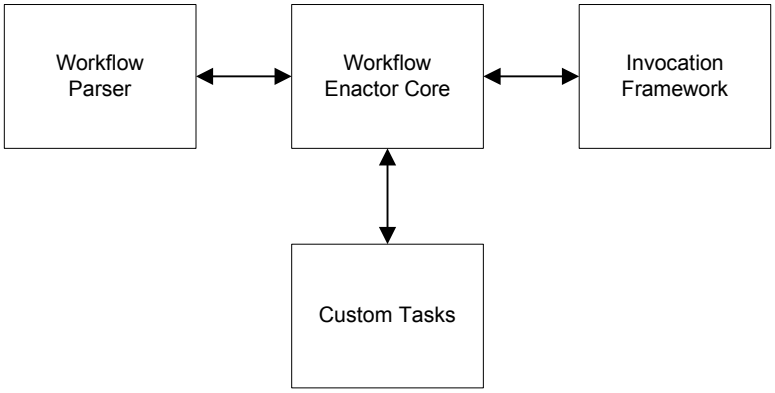


Fig. 8.4 Architecture of IT Innovation Workflow Enactor

8.4.4 Possible Future Projects

8.4.4.1 Newcastle Workflow Engine

The North-East Regional e-Science Centre [NEReSC] is building four Grid services to support their applications with more details in the appendix, section A.2.4.2.1. One of these is workflow.

The ability to capture and enact computations is important for all the e-Science projects. Existing development of a workflow execution engine in the myGrid project (by IT Innovations) is the basis for the Workflow Enactment Grid Service (WEGS) for the Core Grid Middleware. The current myGrid enactment engine is based on Web Services, and so NEReSC is porting it to be a Grid Service.

The WEGS in the initial release of the Core Grid Middleware will accept workflows written in the Web Services Flow Language (WSFL) [WSFL]. However, NEReSC is already evaluating the possibility of providing support for BPEL4WS the Business Process Execution Language for Web Services [BPEL4WS] in addition to WSFL.
NEReSC is planning to follow the work of the GGF working groups in the important area of workflow/business process composition and intends to provide support for the Grid workflow standard when that is available.

The original version of this contribution can be found in the appendix, section A.2.4.2.1.

8.5 Notification Service

8.5.1 Introduction

Notification is an event mechanism for distributed systems in which a notification message is sent to one or more recipients when a particular event occurs. Notification provides a means of asynchronous communication that is distinct from the request/response mode of interaction of the client-server model.

In publish-and-subscribe computing a producer publishes data to a channel to which consumers can subscribe. In general, the data published could be continuous, such as an audio or video feed. However, if data are published only when specific events occur, then we have a notification service with publish-and-subscribe semantics. In the publish-and-subscribe approach to notification, the producer is the active party and the consumers are passive – it is up to the producer to decide on the set of events to which consumers can subscribe, and what data is published when an event occurs. However, some degree of customisation is available to consumers through the application of filters to a channel.

The publish-and-subscribe approach to notification is simple and can be implemented to make efficient use of resources such as network bandwidth. However, it is constraining since it does not allow a consumer to specify the events he or she would like to subscribe to – a consumer can only choose from the set of events specified by the producer.

In the point-to-point approach to notification the middleware maintains a separate queue for each subscription by a consumer. When an event occurs the producer places a data message in the queue, and this message is subsequently received by the consumer. This approach is more flexible than publish-and-subscribe, but can lead to scalability and quality of service problems, particularly when large numbers of consumers subscribe to the same event. Essentially there is a separate channel for each event to which a consumer subscribes. A benefit of point-to-point notification is that notification events and notification messages can be tailored to the requirements of the consumer. This requires the notification source to expose sufficient information to allow a consumer to formulate a subscription request that describes when a notification event occurs, the content of the notification message, where the message should be sent, and the initial lifetime of the subscription.

The notification service in CORBA 3.0 supports filtering and quality of service capabilities [CORBANotification] and is based on publish-and-subscribe semantics.

In OGSI a notification source sends notification messages to a notification sink using special notification portTypes [OGSI]. A subscription request to a notification source specifies when messages should be sent based on changes to values within a service's serviceDataSet, i.e., on its Service Data Elements.

Scalability, quality of service, and authorisation/authentication are important issues.

8.5.2 Summary of Gaps

- OGSI notification service is simplistic and not event driven
  - It is synchronous and suitable for simple events. Seems too simple for traditional synchronous collaboration
  - It will not work for asynchronous notification – dominant use of systems like JMS
- Must support P2P, Publish-subscribe and QoS

8.5.3 Selected Snapshots of Current Activities

8.5.3.1 myGrid and Wrapped JMS (Java Message Service) at Southampton

This service is under development at Southampton University as part of the myGrid project [MyGrid-A]. This *Notification Service* is a service capable of delivering messages in an asynchronous manner.

- It will be standalone Web/Grid Service;
- It will be topic-based and will support both push and pull consumers and publishers [JMS];
- It will be organisable in a peer to peer manner, where a network of notification services is able to route messages between producers and consumers;
- It will support elements of negotiation over ``quality of service'' between consumers and producers.

The original version of this contribution can be found in the appendix, section A.2.5.1.1.

8.5.3.2 Notification Grid Service at Newcastle

The expected dynamic and distributed nature of Grid Applications means that a facility is necessary for informing interested parties of changes in data, Grid Service status, application-specific events, etc. The Notification Grid Service (NGS) is based on a service produced within the myGrid project by the University of Southampton [MyGrid-C]. The myGrid Notification Service provides a way for services to publish events and/or register interest in published events.

As with the workflow enactment engine, the current myGrid Notification Service is based on Web Services and so it is being adapted to be OGSI compliant. The OGSI specification describes a notification portType and Service Data Elements (SDEs) through which NGS will provide the myGrid Notification Service functionality.

The original version of this contribution can be found in the appendix, section A.2.4.2.1.

8.6 Meta-data and Semantic Grid

8.6.1 Introduction

In the simplest terms, metadata is data about other data and is usually intended for consumption and interpretation by machines, rather than by humans. Metadata is aimed at providing machines with the ability to make well-informed decisions about data and processes based on logical inferences. The Semantic Grid is a term used to describe a metadata-rich environment for Grid computing. For a Grid to be transparent and autonomic, and to support communities of users with a range of expertise in a dynamic and heterogeneous environment, requires semantically-enabled infrastructure, and as Grids become increasingly endowed with metadata they will naturally evolve into Semantic Grids.

In a service-oriented architecture for the Grid, metadata is associated with services. For example, a service might contain metadata that describes its capabilities, interfaces, provenance, performance, security and access policies, and so on. This metadata could be used, for example, by a workflow service to decide how two services might interact, or by a resource broker to decide how to schedule a service-based application. The interaction between semantically-enabled services is commonly viewed as being mediated by autonomous software agents. Each service may have several agents that process the service's metadata, negotiate with other agents, and make decisions about the use of the service. These agents usually have access to ontologies and inference engines to support their decision-making processes and to reconcile different XML schema and name spaces.

Recommender systems are a good example of a type of service that makes use of domain-specific knowledge to decide on an appropriate action. Examples of recommender systems are found in mathematics where, given a particular problem, such as the solution of a partial differential equation with given initial and/or boundary conditions, the system recommends a solution algorithm.

Services can also publish their metadata to metadata repositories that can then be queried to discover resources and services that have specified characteristics. MDS or Monitoring and Discovery Services [GlobusMDS] is a set of information service components in the Globus toolkit for publishing and discovering resource status and configuration information. MDS is based on Globus' GIIS (Grid Information Indexing Service [GlobusGIIS]) and GRIS (Grid Resource Information Service [GlobusGRIS]). The information provided by MDS includes current load, CPU configuration, operating system, RAM and virtual memory, free disk space, and network interconnect. Thus, MDS can be viewed as a metadata repository that stores information about computer resource services.

A UDDI (Universal Description, Discovery, and Integration) repository [UDDI-A] stores metadata about the capabilities and interfaces of services. Service metadata is published to UDDI (either manually or by a service deployment tool), and the repository can be queried to discover services that match with specified capabilities and attributes. Query results are returned in the form of URIs that point to the metadata (usually in the form of a WSDL document) of each service satisfying the query.

Meta-data exists at all levels of the Grid from the lowest level repositories of Grid handles to the upper levels defining ontologies and other information about application resources. The overall architecture of these layers of metadata is still debated but it seems likely that a unified approach to meta-data at all levels will be most effective. One may have multiple metadata services corresponding to the different levels but they can usefully share a common repository technology and be designed to support integrated queries.

8.6.2 Summary of Gaps

- Semantic Interoperability for system and application services
    - This is to be addressed by OGSA and there needs to be support of ontologies and rich meta-data systems for entering and searching
- Need to start now generating meta-data and building/deploying tools to aid this
    - Need "personal" (light weight) metadata manager
- Further many "particular technology infrastructure gaps" were in meta-data area reflecting both importance of field and confusion in it
    - Not clear that GGF can resolve this on its own because so much meta-data is discipline specific and no consensus as to how to layer and integrate meta-data from different levels of service
    - Integration of Semantic web and digital library technologies may impact approach
- Adaption of Semantic Web technology to the Grid which has fewer "high value" resources whereas Semantic Web aimed at many more smaller resources
- Development of meta-data registering/look-up service covering MDS (Globus) RGMA (EDG) MCAT (SDSC) Semantic Web and P2P styles and detailed Schema coverage. Support "single service" model (raw UDDI-style service and higher level meta-data in same service) and "hierarchical model" splitting levels into two or more services.
- Evaluate current meta-data tools such as RGMA, UDDI, Spitfire, SRB/MCAT in designing new meta-data infrastructure
- Weakness of UDDI for service registration and discovery
- Data grids need more low level meta-data.
- Need e-Science wide support of provenance technology
- e-Science needs relationships with meta-data activities outside the Grid including ISO, openGIS consortium
- Tools to aid in the mapping between schema and between thesauri will grow in importance as number of (data-rich) application areas increases
- SDE (Service Data elements) in OGSI have no best practice experience and could lead to more inconsistent meta-data specification unless appropriate support infrastructure developed.
    - Need to agree on level of detail in SDE's
- Need to support in discovery different views federation and a range in semantic detail
- Separate semantics and deployment of discovery
- Provenance needs meta-data and tools to process – reason about implications
- Policy driven binding could have implications for meta-data service and its implementation
- SRB could be used here but MCAT is not used in practice for this type of application as SRB's strength is managing meta-data attached to files

8.6.3 Globus Specific Gaps

- Weakness of MDS in GT2 in both technology of implementation and information stored. (? This could reflect confusion in GGF or lack of knowledge of e-Science participants of GGF activities.)
- Missing features of resource descriptions (some of these are present in Unicore) (? This is a problem at GGF of this meta-data responsibility falling in multiple working/research groups)
- Deployment of RGMA to replace MDS

8.6.4 Selected Snapshots of Current Activities

8.6.4.1 UDDI Evaluation

A UDDI (Universal Description, Discovery, and Integration) repository [UDDI-A] stores metadata about the capabilities and interfaces of services. Service metadata is published to UDDI (either manually or by a service deployment tool), and the repository can be queried to discover services that match with specified capabilities and attributes. Query results are returned in the form of URIs that point to the metadata (usually in the form of a WSDL document) of each service satisfying the query.

Metadata about a service needs to be provided in a specific format within the UDDI registry – such as the physical address/location of a service, contact phone number, the category of a service (generally specified based on a Tax Code ID or a Dun and Bradstreet (DUNS) number), etc. Such metadata is therefore specifically aimed at business services, and extending the data structure to register and provide support for services in another domain (such as scientific computing) may lead to incompatibilities. It is also not obvious how such extensions should be undertaken, and whether vendors are likely to support them. Further, search to a UDDI registry is based on a unique key, and capability based search is difficult to undertake. Another significant issue is the management of "top level" UDDI registries – currently undertaken by major vendors, such as IBM, Microsoft etc. Although the vendors indicate that replicated registries are likely to have identical content, there is no obligation for this to be the case. Therefore, there is likely to be inconsistent replication of UDDI registries, which may be further compounded by particular organisations managing their own "private" registry services. Therefore, although the approach adopted in UDDI is a useful one, the current implementations and availability of this technology may be limiting for supporting Grid Services.

Alternative approaches to support registries include the Lightweight Directory Access Protocol (LDAP), which defines a network protocol for querying and updating X500-type directory servers. Although it does not offer a data model rich enough to encode Grid Services, it does have the advantage that many implementations are available (for Linux, Windows, and in programming languages like Java), and the current version of Globus MDS [GlobusMDS] utilises it. The primary factor against the adoption of LDAP is its limited scalability over a distributed environment.

A number of UDP multicast based approaches also provide the capability to support registry services, examples include Jini [Jini], the IEEE SLP (Service Location Protocol [SLP]), and the Service Discovery Service from Berkeley [NinjaSDS]. All of these approaches rely on the use of multicast requests to discover a registry service(s), and therefore are often restricted to a limited part of the network (using a limited number of hops/Time To Live) – and unlikely to scale to the Internet and the Grid. Jini provides a useful approach for enabling Java classes to expose and publish their interfaces with a "look-up" service, along with proxy objects that enable subsequent invocation of the Java classes. The proxy object needs to be downloaded by the client to initiate a session, and can thereby enable a number of different protocols to co-exist. The approach relies on the assumption that all services must define their interfaces using the Java type system, and also that subsequent search for suitable services in the registry (the lookup service) is restricted to these data types. Jini provides the notion of "leasing" (or soft state), which is also now available in OGSA. The Service Location Protocol uses a series of filter expressions to identify the location, types and attributes of a service within a part of the network. The query language is simpler than Jini's. The Service Discovery Service also uses the same principles as SLP and Jini, and support a simple XML-based exact matching query type. The particular advantage offered by SDS is the use of secure channels to discover and publish service metadata. Furthermore, SDS servers can be organised into a hierarchy, enabling an administrator to analyse (and subsequently improve) performance at particular points in the hierarchy. Each SDS node in the hierarchy can hold an aggregated index of the content of its sub-tree.

The original version of this contribution can be found in the appendix, section A.2.6.1.1.

8.6.4.2 Unicore and MDS

Resource brokering is an essential tool for a scalable Grid. There are four functions that a resource broker should be able to perform on behalf of its clients. It should be able to *discover* sites advertising resources, to *check* that the resources are able to accomplish the clients needs, to *negotiate* for quality of service and cost and to *enable* a contract between requestor and client to be agreed. The EuroGrid project is developing a broker that has such functionality building on the powerful abstractions in UNICORE [Unicore]. It is desirable to make this broker interoperable between Globus and UNICORE given the large market share that Globus currently has in Grid computing. In order to accomplish this, an Interoperable Resource Broker is being developed based on the EuroGrid Resource Broker. This work has been carried out at the

University of Manchester as part of the EU 5th Framework Project GRIP IST-2001-32257 [GRIP]. The GRIP broker builds on the functionality established by the EuroGrid Resource broker m[EuroGrid].

The broker searches for resources described via UNICORE to resources controlled by either UNICORE or Globus. The Globus resources are described via MDS schema using the GRIS and GIIS mechanisms. UNICORE has a series of structured abstractions expressed as Java classes that provide a rich vocabulary for describing hardware and software resources used in Grid workflow requests based on an Abstract Job Object (AJO). It also has a workflow language based on the UNICORE protocols (UPL). The interoperable resource broker is being designed to translate between the UNICORE and Globus Resource Description domains. Further details and the architecture design and the implementation are given in the Appendix in Section A.2.6.1.2.

Here we describe the functionality available under UNICORE which is intended to be also implemented on sites running purely Globus. Single tasks can already be passed from UNICORE to Globus. Complex work-flow structures are possible in this framework supported by the UNICORE workflow constructs. These require research into an ontology capable of expressing such resource requests which is much richer than is currently provided via MDS information services. This is beyond the remit of the GRIP project which finishes in Dec 2003, therefore this fuller ontology will be a gap until funding is obtained to develop the ontology work. The EuroGrid broker can broker for UNICORE sites to assemble the following workflow tasks.

- Data pre-processing at site A, simulation at site B, and post-processing at site C.
- Iterative re-execution of a simulation until a termination criterion is met. The simulation may be a complex job, such as in item (1) above.
- Simple application steering, in which a job "holds" itself at some point and waits for the user to check intermediate results before releasing the job.
- More complex application steering, in which jobs are monitored and steered during execution. This may include visualization.
- Conditional execution of tasks, or whole sub jobs, depending on the results of other tasks.
- Ensemble simulation of many cases at different sites and subsequent collation of results centrally.
- Simple meta-computing jobs.

Developing a Grid Resource Ontology has desirable consequences beyond the initial aims of UNICORE-Globus interoperablity. The UK is developing a range of middleware that needs to describe resources and it is highly desirable that mechanisms exist to enable this middleware to interoperate and to adhere to agreed standards. A Grid Resource Ontology is a much more flexible tool for such interoperability than a fixed language for reasons which are described more fully in a document presented to the GPA-RG at GGF7 [Brooke03].

8.6.4.3 Relational Grid Monitoring Architecture

R-GMA (Relational Grid Monitoring Architecture [EDGWP3RGMA]) has been developed in the European DataGrid (EDG) project as a Grid Information and Monitoring System for both the Grid itself and for use by applications. It is based on the GMA from GGF, which is a simple Consumer-Producer model. The special strength of this implementation comes from the power of the relational model. A global view of the information is offered, as if each Virtual Organisation (VO) had one large relational database. A number of different Producer types have been coded. They have different characteristics; for example some support streaming of information. Combined Consumer/Producers are also provided, which are able to combine information and republish it. At the heart of the system is the mediator, which for any query is able to find and connect to the best Producers to do the job. In addition to specific R-GMA sensors able to publish information, tools are available to invoke MDS info-provider scripts and publish the resulting information via R-GMA and also to take R-GMA and publish to an LDAP server.

Work is required to improve the documentation – especially the user guide. Porting to other Unix platforms will be performed to make the code more readily available to a larger set of users and to increase the robustness of R-GMA. Currently distribution is via RPMs which are produced for RedHat [RPM]. Packaging suitable for other platforms should be developed – ideally built from the same generic package

descriptions. Currently the code works without security or with the EDG security module. This should be changed to allow different authentication schemes to be adopted.

The original version of this contribution can be found in the appendix, section A.2.6.1.3.

8.6.4.4 Semantic Grid Work at Southampton

The first explicit characterisation of the Semantic Grid [SemanticGrid] appeared in a report commissioned for the UK e-Science Programme. This work sought to take a service-oriented approach to the Grid and embodied two fundamental assumptions. The first is that an agent-based characterisation of Grid services was likely to be helpful. The second was the importance of having explicit semantic descriptions of the content, processes and services deployed on the grid.

Subsequently work on the Semantic Grid at Southampton can be found embodied in a number of projects. The first of these is GEODISE [GEODISE] in which ideas from the Semantic Web and knowledge engineering are being incorporated into a set of services to support the engineer in design optimisation. Two other projects MIAKT [MIAKT] and CoAKTinG [CoAKTinG] have arisen from the Advanced Knowledge Technologies (AKT) project [AKT] led from Southampton. AKT is developing a wide range of technologies and services for the Semantic Web covering the whole knowledge life cycle - acquisition, modelling, retrieval, reuse, publishing and maintenance.

MIAKT is attempting to develop and apply ontologies as annotations for medical images. It is also uses IRS (Internet Reasoning Services [IRS]) to invoke Grid based processing for image registration (a computationally intensive task of aligning images taken at different times). The IRS takes a high level approach to defining the knowledge level competencies of a service – the sort of processing and problem solving a service is able to perform. In CoAKTinG we are looking to enhance the capability of Access Grids [AccessGrid] and other collaborative videoconference environments. In particular, we are using software that can capture group decision-making and use its outputs as annotations on the different content produced by the meeting.

Southampton is also a partner in myGrid in which a knowledge-oriented view of Grids is adopted in order to discover services and assemble workflows for bioinformaticians.

The original version of this contribution can be found in the appendix, section A.2.6.1.4.

8.6.4.5 SDT Semantic Discovery Toolkit

The SDT Semantic Discovery Toolkit [SDT] is under development at Southampton University as part of the myGrid [myGrid-A] project. The *Service directory toolkit* is a toolkit for deploying service directories in multiple ways:

- All information about published services will be represented in a triple store supporting the RDQL query language;
- Several interfaces will be supported for registering and searching services (including UDDI, JAXR, DAML-S, semantic annotations, biomoby subset);
- Services will be deployable in multiple distributed configurations, including standalone service directory, personalised federation of registries, or tunnelling;
- Service directory will allow third parties to attach annotations to service descriptions (e.g. semantic description, perceived reliability, trust).
- Configurations will be specified by management policies.

The original version of this contribution can be found in the appendix, section A.2.6.1.5.

8.6.4.6 Metadata Management

The original version of this contribution can be found in the appendix, section A.2.6.1.6.

8.6.4.6.1 General Scientific Metadata Format

The Data Management Group of the CCLRC e-Science Centre [UKeS-B] is involved in a range of research projects to develop tools to further the collection of metadata, the integration of different data resources and their exploration potential. The group has over the past two years developed the CCLRC Scientific Metadata Format [Matthews01A]. The Metadata format is currently available in Ontology, XML-Schema and Database implementations and is used for a variety of projects. The metadata is currently used for two different purposes to help to collect metadata at facilities and in projects who have not done so before and as an interchange format (description of information required) with existing facilities. Examples for its application are a number of CCLRC internal departments: ISIS, SRD (test data only), as well as a range of UK e-Science projects: a Polymorphism metadata database (Sally Price, UCL, e-Materials project), a Combinatorial Chemistry metadata database (Richard Catlow, RI, e-Materials project) and a set of Surface and Surface Fluid Interaction metadata databases for the NERC e-Minerals Project (Martin Dove, Cambridge, Richard Catlow, RI, David Price, UCL, Stephen Parker, Bath,). Otherwise the CCLRC Scientific Metadata Format is used to interchange information between existing sites within the DataPortal tool [CCLRCeSCDP-A].

8.6.4.6.2 CCLRC Data Portal

The Data Management group of the CCLRC e-Science Centre has developed a cross-disciplinary DataPortal [CCLRCeSCDP-A], which allows searching various data facilities in parallel, exploring their metadata catalogues and accessing their data. It was important for us to preserve the independence of existing data facilities, therefore the Portal's work does not require any specific database technology, metadata schema locally or user administration (that is it has to be done electronically so that linkage to certificates is possible). We are currently using XML Wrappers to 'translate' Portal queries into local formats and to generate the replies to the portal. In the future it is hoped that we can make use of ontology mapping services instead as well as a more universal XML query language. To be able to formulate queries and to collate results, we have developed a General Scientific Metadata Format, which is used by the system. The DataPortal has recently been released in its 3rd version [CCLRCeSCDP-A] and is now based on web services technologies (WSDL, Apache's Axis SOAP, UDDI), using UK e-Science Certificates for authentication (for a trial access go to [CCLRCeSCDP-B]). The code consists of a variety of independent modules with web services interfaces based around major functionalities e.g. shopping cart, authentication and authorisation. The Portals security system is tied in with the existing security measures at each of the data facility sites. The user is e.g. able to pose queries, explore the metadata from various sites, access the data itself, save search results in a permanent shopping cart and transfer data to other systems or services. The CCLRC implementation of the Portal currently links three CCLRC test data repositories (ISIS, BADC and SRD) and one external site (MPIM in Germany). More operational catalogues will be added over the coming months. Furthermore, different instances of the Portal are being set up for other projects, such as the e-Minerals Mini Grid [eMinerals] and the e-Materials project [eMaterials]. The Portal's web services technology allows easy linkage to other web services such as they are provided e.g. by the CCLRC HPCPortal [CCLRCeSCHPC]. In the future we will encourage local data sites to register their data with local SRB servers [SRBMCAT] if that is appropriate, so that the metadata will no longer contain hard to maintain physical links to the data, but instead incorporates 'soft links' to SRB logical names.

8.6.5 Possible Future Projects

8.6.5.1 Semantic Grid Expectations

Future developments of the Semantic Grid [SemanticGrid] are inextricably bound up with those of the Semantic Web [SemanticWeb]. For example, it is already possible for grid developers to exploit RDF standards and tools. Moreover, the W3C efforts towards an ontology web language [OWL] are important for the Semantic Grid too. It is to be expected that the following will be key areas for development:

Ontologies – there will be a central role for the use of shared conceptualisations for both the structural content and processes that will be supported on the Grid. Thus, heterogeneous and distributed data

integration and database access will be mediated through ontologies; ontologies will determine workflow composition and also facilitate service discovery and composition.

Annotation - the use of ontologies will call for tools that can carry out large-scale annotations of content with the metadata that these ontologies represent.

Maintenance – the use of extended annotations and ontologies will carry with it significant maintenance overheads. Ontologies have to be managed and maintained. In some areas of science and engineering the concepts are evolving and changing at a dramatic rate requiring new kinds of annotation to be incorporated into metadata repositories.

Intelligent Brokering and Service Discovery – this will increasingly feature as services are composed on demand. There could be considerable scope for negotiation between services. For example, as trade offs are considered with respect to features such as speed of delivery, cost, numbers of completed runs, amounts of redundancy etc.

The original version of this contribution can be found in the appendix, section A.2.6.2.1.

8.6.5.2 A Semantic Grid Service Registry from IT Innovation

One feature of the OGSA standard [OGSA] is that it supports a factory/instance model for services, in which users can create their own service instances to handle their service invocation requests and maintain any state required by the application. A user must first locate a suitable factory and create their service instance, on which they can then invoke operations.

However, the "native" invocation model of the web is "session" oriented – one retains a session identifier (possibly stored in a cookie), and uses this when invoking remote services. In fact, the OGSA standard does not preclude this model – it is likely that factory services will need to use it.

IT Innovation [ITInnovation] plans to propose a new "semantic grid" project in which the distinctions between these two models are abstracted by the invocation framework (extending the work described in section 8.2). The service registry would support semantic searches based on service capability independent of implementation features. In principle this would allow applications to be developed that can use services from different grids, regardless of whether they are deployed in a factory/instance or a session-based grid-hosting infrastructure. A key challenge for this work is to define standards for service descriptions and Grid service registry federation, such that a Grid user can discover the correct starting point from which to invoke a service independently of whether it might involve a factory.

The original version of this contribution can be found in the appendix, section A.2.6.2.2.

8.6.5.3 Provenance Specification

myGrid will offer some provenance support [MyGrid-D], by recording what is being executed and when. This myGrid work will focus principally on the type of data (and their ontologies). The solution will be simple and centred on the enactment engine publishing such an information into a notification service, with a consumer storing it in a repository. In reality, a solution is required that is distributed, scalable, and secure, not only for registering provenance data but also for reasoning about it. This is a new research area in itself: it requires us to design new algorithms for submitting provenance data asynchronously and securely, to study their correctness, and to design high performance services able to execute them.

The original version of this contribution can be found in the appendix, section A.2.6.2.3.

8.6.5.4 SRB Evaluation at CCLRC e-Science Centre

Storage Resource Broker (SRB) is a client-server based middle-ware [SRBMCAT] initially developed by SDSC in the mid-Nineties to provide uniform access interface to different types of storage devices. SRB provides an uniform API that can be used to connect to heterogeneous resources that may be distributed

and access data sets that may be replicated. SRB is being investigated at the CCLRC e-Science Centre as a means of allowing users to manage data storage and replication across the wide range of physical storage system types and locations available within UK e-Science, while still allowing having a single, stable, access point to the data. Other systems, such as RLS from EDG [EDGWP2RLS] and Replica Management [GlobusReplMan] from Globus, are being developed, but SRB is the only currently available and proven tool with the comprehensive set of features required.

It has two major components, the core SRB, which interfaces with the storage devices, and the MCAT, which holds the metadata elements. Many different platforms and authentication methods are supported by the modular design, and a web service interface is available.

The system provides interfaces for the ingestion of data and associated metadata; management of replication and data movement; searching the metadata for discovery; and the retrieval of the data itself. Metadata held to support these interfaces includes the physical and logical details of the data held and its replicas, user information, and security rights and access control. The CCLRC e-Science Centre has agreed a collaboration plan with SDSC to jointly develop any additional functionality necessary for the UK e-Science environment. This will include developments to:

- allow a distributed and metadata infrastructure supporting multiple federated SRB systems (initial implementation in the next SRB version in August 2003);
- disjoint security management for the data and metadata, and integration with the developing UK e-Science security infrastructure;
- enhancement of the SRB security model to include the VO models being developed to support e-Science.

In addition and if required to meet UK needs, we will be looking at extending the currently supported platforms for data and metadata storage.

The original version of this contribution can be found in the appendix, section A.2.6.2.4 with background in section A.2.7.2.3.

8.7 Information Grid Technology including OGSA-DAI

8.7.1 Introduction

A hallmark of the e-Science program is its emphasis on Information Grid projects. These are usually called data grids but this nomenclature is also used for Grids dominated by file access. The technology centerpiece is OGSA-DAI which is defining the Grid services involved in integrating a variety (XML or relational) of databases with the Grid. This e-Science project has made impressive progress and made an early release of both software and specifications [OGSA-DAI]. The SDSC SRB-MCAT project [SRBMCAT] also supports large data repositories but so far has not been adopted in the UK because of difficulties in deployment and its focus on file systems and not databases for repositories. Obviously Grid enabled databases are critical for e-Business but several major e-Science areas rely on them. Most advanced here is the Bioinformatics community that has developed multiple distributed databases and a significant service infrastructure to curate and deliver the data to academic and business clients. However the virtual observatory projects have already demonstrated similar systems while earth and environmental science fields have similar challenges. There is no single solution as the data varies from detailed (high-value and relatively low-volume) curated records to large image datasets annotated with meta-data stored in databases. Both the OGSA-DAI developers and users identified several new features needed in this standard. One important capability is the need to allow filters (often today invoked in Perl for Bioinformatics) so that the Grid service does not directly expose the database (in XQuery/JDBC style) but rather the result of running the filter on the database. This support can get complicated if one needs to support user defined filters and resource brokering on the computing subsystem used to run the filter. Other areas extend well-studied database issues to the Grid – here we can identify federation, multiple versions of Schema, rollback (history) and complex views. Security includes support for confidential access so proprietary secrets cannot be inferred from study of the database access patterns.

Capitol Radio has a requirement to support distributed delivery of multimedia files to support a network of affiliate radio stations. This has rich meta-data but the heart of the problem is the multimedia files which need a peer-to-peer style delivery mechanism that provides documented delivery; for example an advertisement could only be charged if it is both delivered and played. This problem has in fact some overlap with the replica management problem described for particle physics in the next section. However the unstructured real-time nature of the media problem and its peer-to-peer structure differentiate it from the much larger but more structured particle physics case.

8.7.2 Summary of Gaps

- Extensions to OGSA-DAI to support federation, on-the-fly filtering
- This can be broadened to set of domain specific interfaces OGSA-DAI-AstroGrid, OGSA-DAI-EBI, OGSA-DAI-Environment etc. We need to analyze the common features of DataGrid application areas and build domain specific support on top of a more advanced Grid Service framework for collections of Scientific data repositories. Capabilities needed include
    - Federation & Integration
    - Complex Views
    - Schema versioning
    - Rollback
    - Computing on demand for data rich problems
    - Security
- Capitol Radio requirements
- Data curation needs substantial attention (Appendix section A.2.9.2.2(e) and [Curation-A] [Macdonald02A])
- Need to support existing meta-data catalogs in a variety of fields that will NOT change as well established. Requires wrapping
    - Have both cases of collocated and separated data and meta-data

8.7.3 Current OGSA-DAI

OGSA-DAI is an 18 month project (appendix sections A.2.7.1 and A.2.9.2.2(d) [OGSA-DAI]) from February 2002 to July 2003 supported by the UK e-Science Core Programme, through the National e-Science Centre [UKeS-C] and the regional e-Science Centres at Manchester [ESNW] and Newcastle [NEReSC], with IBM and Oracle as industrial partners. The OGSA-DAI project has produced and released the following grid services:

1. Grid Data Service (RDBMS and XMLDB)
2. Grid Data Service Factory (GDSF)
3. Grid Data Service Registry

These are (almost entirely) database agnostic. Essentially, the "front" of the GDS is the same for relational and XML databases, and to add support for a new database someone would essentially need to write a new configuration file which provides a binding to that database.

In theory, no code needs to be changed to support e.g. DB/2 or Oracle as opposed to MySQL - in reality there needs to be minor modifications due to the way that the "standard" URI scheme for opening a connection to a database is not quite as standard as we'd have liked. GDSF is completely database agnostic, except where you choose to include support for any proprietary database management functionality.

In essence, the OGSA-DAI project provides a collection of functionalities for registering, creating and using Grid Data Services (GDSs). The clients of a GDS are able to access and manipulate the contents of a relational or an XML database by way of a GDS instance. Once a GDS instance has been created or discovered, there are in general three parts to an interaction of a requester with a GDS, as illustrated in figure 8.5. In the first part, the requester uses the GridService portType of OGSI (e.g., by way of the FindServiceData operation) to access metadata on the service (e.g., relating to the schema of the database). If the requester already knows enough about the service to use it, this part can be omitted. In the second part, the perform operation of the GridDataService portType is used to convey a request to the GDS, for

example, to evaluate a query. The results of the query can be returned to the requester directly, or to a third party. In the optional third part, several GridDataServices may exchange messages by way of the GridDataTransport portType with other Grid Services.

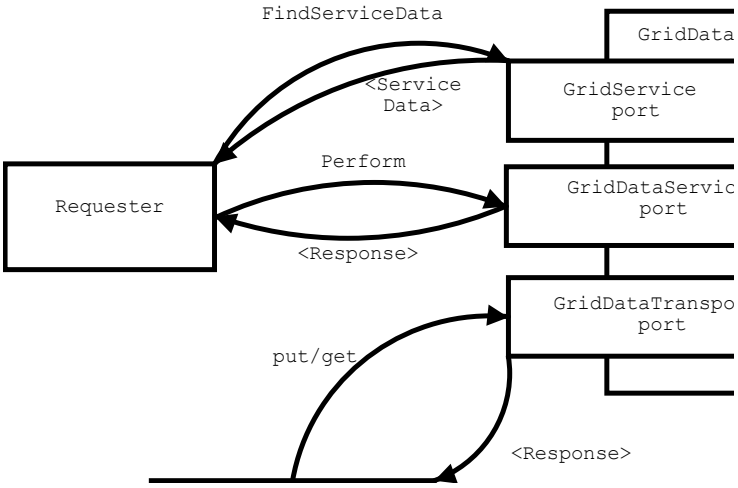


Fig. 8.5: Interconnection of OGSA-DAI Components

The OGSA-DAI software supports the mySQL, DB2 and Oracle relational databases and the Xindice XML repository. The OGSA-DAI project is also providing input to, and contributing a reference implementation for, the ongoing activity within the Database Access and Integration Services Working Group of the Global Grid Forum. OGSA-DAI software is available from [OGSA-DAI]

As well as the development of core data access services, OGSA-DAI will deliver prototype software that demonstrates the use of distributed query processing techniques for integrating data from multiple GDSs. This prototype software will also provide combined use of GDS and other Grid Services, for example by allowing the results of a query over multiple GDSs to be conveyed to an analysis service, and for the results of the analysis to be related to values accessed through further distributed GDSs.

eDIKT described in the appendix section A.7.4 is extending OGSA-DAI for data stored in filesystems and different hosting environments [eDIKT].

8.7.3.1 North-East e-Science Centre OGSA-DAI Activities

The North-East Regional e-Science Centre [NEReSC] is building four Grid services to support their applications with more details in the appendix, section A.2.7.1.3. These services are based on existing e-Science work and this is case for the two described here.

1. **OGSA-DAI**
   Many of the projects at NEReSC require access to database management systems over the Grid. The Open Grid Services Architecture – Database Access and Integration (OGSA-DAI) service provides a consistent way to access relational and XML data on the Grid. The OGSA-DAI implementation, discussed above is therefore included in the NEReSC Core Grid Middleware.
2. **OGSA-DAI Distributed Query Processing**
   NEReSC is directly involved in the design and implementation of the OGSA-DAI Distributed Query Processing (DQP) Grid Service in Sec. 8.7.5.2, which enables Grid applications to run queries over distributed data resources. A number of NEReSC e-Science research projects will use this service to federate data from multiple databases over the Grid.

8.7.4 Selected Snapshots of Current Activities

8.7.4.1 Environmental Grids

There are a number of initiatives both nationally and internationally in the environmental sciences, here we concentrate on grids in the atmospheric, oceanographic, and earth observation areas.

The NERC DataGrid: The NERC DataGrid is a UK NERC/national core e-science project [NERCDataGrid] that began in September 2002. The main aim is to build a grid that connects data collections held by individual scientists with the British Oceanographic and Atmospheric Data centres, and develop the technology to extend that grid to link into the other environmental disciplines for which NERC holds curated data. Clearly work is at an early stage, and at the moment the project is concentrating on developing: (i) a schema which is capable of being used as intermediate schema between the individual highly specialised schema typical of the disciplines involve (e.g. oceanography and atmospheric sciences); and (ii), a simplified data model which can support observational and simulated data in existing formats and using existing transport mechanisms (including OPeNDAP or the Open-source Project for a Network Data Access [OPeNDAP]). The current plan is to use SRB [SRBMCAT] to manage data held amongst the larger centres and a specialised grid-service (based on OPenDAPg [OPenDAPg] developed by ESG [ESG], discussed below) for data delivery to the wider community. It is not yet clear whether OGSA/DAI (section 8.7.3) will play a role in querying databases, or an approach based on OAI harvesting [OAI] of metadata will be used, or a combination of both. Additional work is also being carried out on editors to make it easier for data providers to create the extra metadata required for the grid technologies to be useful, on conventions and standards for the content to go into the schema, and on tools to manipulate data in the grid environment.

The US Earth Systems Grid (ESG II): Following on from ESGI, this DoE funded US project [ESG] began in 2002 and is aimed at making the TB of data produced by climate simulations available to users throughout the continental US. Current testbeds demonstrate the integration of a wide range of Globus based technologies with tools to extract data from deep storage and move it round the country. One key piece of work has involved integrating GridFTP [GlobusGridFTP] into the OPeNDAP transport system so that OpeNDAP servers can utilise fast reliable secure data services (OPenDAPg). Further work on metadata systems is underway including both the development of simplified schema (as in the NERC DataGrid) and potentially on using OGSA/DAI to find and utilise that metadata. They are also working on the use and extension of globus tools to provide community authentication and rights management.
Client software for the ESG is being based on portals, and both fat- and thin-client technologies.

EU DataGrid includes work package 9 on Earth Observation data [EDGWP9]: here the application is to link data processing of EU earth science data in different national institutions (primarily Italy, France and the Netherlands) using the EU DataGrid infrastructure.

The original version of this contribution can be found in the appendix, section A.2.7.2.1.

8.7.4.2 AstroGrid

The AstroGrid project [AstroGrid] intends to develop the world's first implementation of a Virtual Observatory (VO) in which hundreds of data archives and astronomical tools are made available to astronomers via standards-based desktop and web applications.

The project has just now (early 2003) entered its build phase with a fixed completion date of December 2004. In order to develop working services the project will initially develop components as web services, refactoring them as grid services in the second half of the build while extending the functionality to make use of advanced grid technologies.

AstroGrid has defined a high-level model of VO components, of which the key ones include a *Portal*, *Registry*, *Community*, *Data Centre and Dataset Access*, *MySpace* and *Logging*. These are described in detail below

The interface between each of these components will require standards to be defined, many of them in an effort by the international astronomical community (via the iVOA – International Virtual Observatory Alliance [iVOA] – in which AstroGrid plays a leading role).

There are substantial challenges in this project to those developing basic Grid and Data Grid services. The *Portal* component must be able to adopt the *identity* of a user so that jobs can be submitted, data accessed, resources utilised, all without the user having to reconfirm their identity. The *Registry* must contain sufficient metadata about datasets that users can identify ones which have a high probability of being able to satisfy their queries, before those queries are submitted. A *Community* will contain users and groups of users with widely varying rights of access over data; for instance, a subset of data might be only available to one astronomer but that person might want to delegate some access rights to certain colleagues – and after two years the data reverts to being available to all those with rights of access to the dataset as a whole.

Dataset access provides the greatest challenge. Data might be held in relational, object-oriented, xml-based or any other kind of database; it might be in flat files of many different formats: both ascii and binary; it might represent tabular, image, spectral or other types of data. The VO must make all of this data accessible via a standard query language which recognises the type of data involved but allows interpolation so that a query might refer to a data item which must be calculated from other items in the actual dataset.

MySpace perhaps offers an area of development which will prove most useful to other projects. The concept is of a virtual *directory space* of data items, located in databases or files anywhere on the Grid, but which the user can see and manipulate through an Explorer-type interface. AstroGrid has adapted this idea already to cover both *data centre* cache provision and community-based private storage. This is a facility which users in many fields will want available and is one which could be a standard feature of any data grid.

Another common tool that all projects would wish to deploy is a *logging* facility. The ability to log events with custom identifiers is common to many operating systems and should also be a standard feature of the grid.

The original version of this contribution can be found in the appendix, section A.2.7.2.2.

8.7.5 Possible Future Projects

8.7.5.1 Capital Radio

Capital Radio is the UK's leading commercial radio group [CapitalRadio], with greater revenues and profits than any other commercial radio company. This is achieved through a total of 20 analogue radio licences broadcasting to over half of the UK's adult population. These 20 analogue licenses have a near one-to-one correspondence with physical radio studios located across the length and breadth of the UK. Although Capital Radio Group has a large number of separate offices within the country, each radio station has operations that are both centrally and locally directed. There are specific sites that perform specialised functions for the group where their actions can affect the behaviour of sub-group of offices. For example, the London office is responsible for sales of National advertising campaigns. These sales affect the commercial airtime of local radio stations. Another example would be of a single radio station promoting a new artist or new material that is picked up by other stations. These are two simple but common examples that illustrate that the flow of information around the group is multidirectional. A central server solution is inappropriate due to quality of service requirements. First, the stations need to be able to operate in isolation even in the event of network or server failures. Secondly, many data transfers are time critical that could not be met by a centralised architecture in peak conditions. These requirements to have collaborating islands of networks has lead to the consideration of data grid technologies to solve these problems.

The original version of this contribution can be found in the appendix, section A.2.7.3.1.

8.7.5.2 Futures of OGSA-DAI

Current funding for OGSA-DAI ends in July 2003. The current OGSA-DAI project [OGSA-DAI] will produce database access services for use with relational databases and XML repositories. A proposal [OGSA-DAIT] for a follow-on project has been approved by TAG, and this will seek:

- To extend the functionality of the OGSA-DAI Grid Database Services (GDS), for example, to support a richer collection of data movement mechanisms, more comprehensive transaction management, more flexible data transformation, more comprehensive metadata, and more fully integrated notification support.
- To improve performance and dependability, for example to exploit emerging data movement services and transport bindings, to exploit replicated data or materialised views, and to improve scheduling and resource allocation for complex requests.
- To develop higher-level services and application patterns, for example, by developing the prototype distributed query processor into a supported middleware component, by providing facilities to support selective replication of GDS managed data, and by supporting archive management using GDSs.

As the GGF standardisation process for OGSI [OGSI], OGSA [OGSA] and DAIS [OGSA-DAIS] will all continue well beyond the end of the current OGSA-DAI project [OGSA-DAI], the follow-on [OGSA-DAIT] would seek to track and inform standards in various parts of the GGF, provide a reference implementation for the DAIS standard as it evolves during the GGF process, and inform future versions of the DAIS standard. It seems likely that many GGF standards will have multiple versions, and thus that the DAIS activity will not only have to evolve to stay in line with other standards, but also adapt to make use of developments elsewhere in the Grid community.

The original version of this contribution can be found in the appendix, section A.2.7.3.2.

8.7.5.3 Environmental DataGrid

Datagrids, in common with other projects, are grappling with rights management issues. While there is much going on to deal with security of data transfer and Globus offers tools for secure access to resources, there are no existing reliable tools for matching users (with defined roles and rights) to data elements (with restrictions and access policies). These issue will become very important as there is both commercial sensitivity and intellectual property issues associated with some data products, both of which need to be addressed, before datagrids will be in common use.

All datagrids are also grappling with issues associated with metadata schema and how to handle large volumes of existing metadata only some of which is encoded in consistent well-understood ways. Methods to evolve the metadata content into ISO standard compliant metadata is being compounded by the shear number of the relevant ISO standards and their obscurity. No current datagrid research project has the resources to deal with even keeping track of ISO issues, let alone being fully compliant. In addition, the issue of how to bring convergence between the Open GIS consortium (OGC) concepts of web-services [OGCWS], and the OGSA grid-services will also become important as both of these become more common.

There is a necessity to address methods to automatically generate commonly understood schema from one or more individual schema: for example, where two sites have rich semantics encapsulated in different database or XML schema, only some of the attributes will be understood in common, but it is these for which distributed queries will be understood. Techniques to make the process of identifying commonly understood attributes from multiple schemas and dynamically generating a consistent query language with appropriate rules will considerably aid in what can be done in the development of inter-disciplinary datagrids.

There are a number of initiatives to look at inter-operation of datagrids. In particular, the NERC DataGrid [NERCDataGrid] and the Earth System Grid [ESG] intend to deliver inter-operation of metadata and data transfer tools during 2004. This interoperation will no doubt stress elements of both grids and the international network links (for example, ESG I demos have shown sustained data delivery bandwidths of

over 200 GB/hour, such bandwidths are the minimum necessary if these grids are not to be anything more than glorified file-transfer grids). Other significant interactions include those brokered by CEOS (the Committee for Earth Observation Satellites [CEOS]) who hope to deliver interoperation between the Earth Observation community and the climate modelling community (ESG) as a testbed to demonstrate what can be done in this area; and those that might be engendered should the EU FW6 project CAPRI (Coordinated Access to the PRISM Infrastructure) be funded. While PRISM, the programme for Integrated Earth System Model [PRISM] is not badged as a grid project, it shares many common characteristics, and while it is based mainly on distributed computing issues associated with climate modelling, they have perceived a necessity for a distributed data management system that is likely to be grid based. This would be a significant component of the CAPRI project, and again, interoperation with ESG is seen as crucial.

The original version of this contribution can be found in the appendix, section A.2.7.3.3.

8.7.5.4 AstroGrid

The AstroGrid project [AstroGrid] is already considering how it might extend the Virtual Observatory (VO) beyond the facilities considered in the section 8.7.4.2 above.

Several of the researchers involved in the project are already considering how the VO might be extended into the Semantic Grid. The nature of data in astronomy is relatively complex with consequential complexity in the semantics of its metadata. One of the key challenges is to describe metadata relationships in a way that recognises that the relationship might be problematic: for example, that only 10% of astronomers hold that the relationship is true. The project would wish to evolve a query language and metadata registry which allowed such complexities to be taken into account.
It is important to emphasise that existing commercial component models such as EJB and the CORBA Component Model are *not* what we are talking about. These are used to build *applications only*; they are too heavyweight and prescriptive to implement systems themselves. The same applies to the Grid-oriented component models that have so far been developed, e.g. ICENI from Imperial College [ICENI].

Astronomers already rely on a large number of tools, most of them used currently on their desktop machines. The project will seek to make these tools available on the grid such that they can be deployed at data centres alongside the dataset of interest with the user unaware of any movement of data or executables.

Mining the massive datasets which new missions are producing and will produce in the next 10-20 years is a critical activity and is essential to the future of the VO. Tools must be developed to mine such data using techniques which are efficient on the huge volumes, can recognise patterns and ambiguities and which can make use of the semantic content of previous queries to *suggest* approaches the user may not have thought of.

In all future extensions to the VO, it is essential that grid standards and VO standards evolve to complement each other. This *social* challenge is perhaps one which will tax the projects and people involved most.

The original version of this contribution can be found in the appendix, section A.2.7.3.4.

8.8 Compute/File Grids: Scheduling, Access to Mass Storage, Replica Management and Virtual Data

8.8.1 Introduction

The Grid work initially focused on integrating compute resources with their associated data repositories. The two largest projects outside the UK have a major focus in this area – these are the US Trillium effort (GriPhyN, iVDGL and PPDG [Trillium]) and the EDG (European Data Grid [EDG-C]). Note the confusing feature that we classify EDG as a compute Grid activity because it and Trillium have an early focus on the batch analysis of particle physics data. This field generates the most data of any current application – hundreds of petabytes are expected from the LHC [LHC], but this data is more structured than the lower

volume database oriented applications we have put into the Data Grid category. GriPhyN [GriPhyN] through the VDT (Virtual Data Toolkit [VDT]) has put substantial effort in hardening core software and their latest release (which includes some EDG contribution) includes both GT2 [Globus-B] and Condor [Condor]. It is expected that this software will be augmented by work funded by the US NMI (NSF Middleware Initiative [NMI]) project. EDG also has major software efforts in this area which could be of great value to the e-Science effort. As described later, the Rutherford Appleton Laboratory RAL has identified EDG replica, mass storage interface and resource broker software as major compute Grid software capabilities that could be generalized from the EDG work and used in the UK e-Science program. The EDG resource broker operates at the planning level (matching of job requirements to compute/data/network capabilities) and not the execution level. The meta-data section notes that EDG currently supports a richer job model than the VDT reflecting their adoption of ideas from UNICORE [Unicore-A]. We can expect international collaboration to reconcile these differences but the timing and emphasis of such merging of core capabilities could possibly be accelerated.
The VDT currently includes Chimera - a "virtual data" tool - [Chimera] which is a major focus of the GriPhyn project. Virtual data refers to the concept that data can be referenced in a job not just in terms of a handle to its storage location but rather by a "process" that defines how it can be generated – either from a nearby cache or from a script that runs to generate it from other data. This latter script is specified in some language closely related to that used in workflow. Virtual data was not highlighted as important in the interviews but workflow described earlier was a major emphasis and we can expect workflow and virtual data research to merge.

The White Rose Campus Grid [WhiteRose] which can be expected to be replicated in other organizations clearly makes extensive use of this area of Grid software. They also need broad based support for base technology and support of multiple resource brokers; White Rose uses the Sun Grid Engine [SGE] rather than Condor [Condor], which was adopted by the VDT.

The Information Grid projects were currently focused on issues of accessing, integrating and viewing information stored in databases. However integration of Information and Compute Grids will grow in importance as these projects move to stages where information from databases needs to be passed through user or system filters. The particle physics effort (through replica management) has addressed carefully the issues of collocation of data and computer resources; do you move computing to the data or vide versa? They have designed a multi-tier computing model with central (CERN), national, regional and local resources. The analogous architecture for other applications should be studied; these might have fewer available resources, different compute/data bandwidth ratio and different security/proprietorship constraints on the data.

8.8.2 Summary of Gaps

- It is not clear if current Grid technology suites are robust enough for the needs of particle physics
- Generalize replica management outside particle physics and current Globus/EDG implementations
- Could need broad support of virtual data or of the simpler replica management capabilities as being developed by DiscoveryNet, GriPhyn, EDG
- The area of scheduling covered by Condor and commercial tools like the Sun Grid Engine and LSF from Platform Computing was discussed but major gaps were not identified
- Integration of mass storage with Grid major effort in EDG
- It was noted that problems arise with such schedulers if a given resource is part of multiple grids and could be allocated by multiple schedulers. This issue has been noted in the design of the proposed Grid infrastructure to be developed by the UK Open Middleware Infrastructure Institute (see Section 13).
- Interestingly Compute/File Grids which are stressed in major US Grid projects like GriPhyn were not discussed significantly outside particle physics and the Campus Grid of the White Rose. However this reflects partly that compute grids are more mature than information grids.
- Globus replica catalog [GlobusReplCat] centralized with one server per VO; need more distributed solution
- Daresbury considers SRB [SRBMCAT] "only" game in town for managing meta-data for distributed files and replicas – EDG and Globus make a different choice
- MCAT part of SRB needs to made distributed and support additional databases

- UNICORE job submission attractive [Unicore].

8.8.3 Possible Future Projects

8.8.3.1 Hardening of EDG Replica Technology

The Replica Management Service [EDGWP2RMS] is being developed in the European DataGrid (EDG) project as a generic solution to file-based Grid data management. It provides a single transactional interface [EDGWP2RLS] to four sub-services: the *Replica Location Service* (developed jointly with the Globus project) which keeps track of distributed file replicas; the *Replica Metadata Catalogue* which permits applications to store meta-data relevant to their own files; the *Replica Optimization Service* which uses information about the available Grid resources to make decisions about when and where to create new replicas; and the *Replica Storage Handler* which provides subscription-based replication, and provides the interfaces to the Grid services which move the data.

Work is required to improve the documentation (with emphasis upon installation and configuration), to undertake thorough testing in different environments and to improve the packaging of the software. These are the key areas that are required to make the product easily accessible to a wider community. By doing this in close collaboration with the original developers, the appropriate support channels will be set up, and the necessary experience will be gained in the Grid Support Centre for the longer term support of the software.

Note that the EDG RLS system is potentially more attractive than the implementation of RLS which will be distributed with Globus 2.4 [GlobusReplMan] as it uses Web Services communication protocols rather than proprietary Globus protocols and so can be used independently of Globus. Schema changes and addition of different databases are also easier.

The original version of this material can be found in the appendix section A.2.8.1.1.

8.8.3.2 Hardening of EDG Compute Resource Broker

The Resource Broker (RB) [EDGWP1Arch] [EDGWP1-B] being developed in the European Data Grid (EDG) project [EDG-A] is central to the EDG middleware. It is this component that matches the requirements of a job to the best available resources. In this decision it analyses the data needs of the job (by interrogating the data services) and then compares the other requirements and preferences as expressed in ClassAd [Condor-ClassAds] form by the user submitting the job. The development of the RB has been focused outside the UK (mainly in Italy), however the UK has been involved in debugging, testing and quality assurance for the RB. Several other projects have expressed interest in the functionality of the RB and the work being performed to make it OGSA-compliant will enhance its reusability in other projects.

A repackaging of RB that makes it more of a standalone product and easier to install would enhance its re-usability in other projects. Currently the RB only works on Linux, the porting of the product to other platforms (in particular Solaris) would greatly enhance its reusability and would naturally accompany the repackaging. The greatest enhancement to the reusability of the RB in other projects comes from the work being done to make the RB OGSA-compliant. This work will require documentation, packaging and a distribution mechanism via the Grid Support Centre. The core RB development will continue to be performed outside the UK, however the UK has very good relations with those developing the core development and Web Services wrappers are being developed in the UK.

The original version of this material can be found in the appendix section A.2.8.1.2.

8.8.3.3 Hardening EDG Storage Element - Grid interface to mass storage resources

The StorageElement (SE) being developed [EDGWP5] in the European DataGrid (EDG) project is a Grid interface to Mass Storage Systems (MSS). It allows authenticated access over the wide area network to data on disk and/or tape robot systems using a variety of protocols. The SE can be used directly by end users or

indirectly through the EDG replica manager service. It is probably of most interest to sites with MSS although it is also implemented on simple disk-only systems.
The SE has three logical interfaces. The innovative one is the Control Interface which allows users to reserve space for writing data or to request staging of files from tape to disk in anticipation of subsequent transfer or local access. When a request is made, the SE returns a filehandle in the form of a Transfer URL (or TURL) which specifies where the relevant data should be read or written and the protocol to be used. The Data Interface supports the relevant protocol (GridFTP in the first release [GlobusGridFTP]) to move data in and out of the SE. The Control Interface uses the relevant Grid Information Service (MDS now [GlobusMDS], R-GMA later [EDGWP3RGMA]) to publish metadata about the SE and its files to enable jobs to be scheduled on services with access. The SE uses the same authentication and authorisation as other EDG software but also applies file-level GACL to give a finer-grained control over access. The control interface is implemented as a C API or as a web service via a java API. A command line interface has been implemented using the C API. The core SE system makes extensive use of XML for metadata, control, and execution. The modular design of this core allows the interface to a specific MSS to be concentrated in a few places within the code allowing straightforward porting to new MSS.
The SE is under development and is being integrated with EDG Testbed 2.0 in March and April 2003. This first release will support MSS at CLRC and CERN and use GridFTP and MDS. Work to be done includes: identifying the MSS of interest to UK data archiving and curation sites; porting the SE software to their preferred operating systems; writing handlers for these MSS; developing support for additional data transfer protocols like HTTP to reduce reliance on Globus; and adding group-based quota handling to support the mixed user environments of university computing centres.

The original version of this material can be found in the appendix section A.2.8.1.3.

8.9 Other Technology areas

8.9.1 Introduction

This section contains a set of areas which although important did not directly fit in the major Grid components discussed earlier; some are placed here because they were common to both Compute and Data Grids and so did not fit in these last two subsections. The area of fabric management is already a major UK activity with LCFG from Edinburgh [LCFG] the basis of EDG work in this area and GridWeaver [GridWeaver] from HP addressing features of future systems. These systems automate the configuration of large clusters and obviously such fabric support is essential if Grid resources are to be reliably managed and further both initial and dynamic software deployment should be linked to the meta-data catalogs (discussed in section 7.6 and 8.6) and higher-level services such as Gridmake (section 8.9.4.2).

Accounting and the still experimental but intriguing Grid economies are other critical services that are needed for all types of Grids. The UK has deployed the Access Grid widely and integrating collaboration with the Grid in a broader fashion is being pursued but not so far in a coordinated fashion.

Visualization has always been difficult to standardize but it would certainly be important if progress could be made in agreeing on architectural principles for a visualization Grid service. There is now some experience in using both distributed object and XML specified data structures in visualization. Further the UK has set up a visualization community and it would be interesting if these steps could lead to progress in visualization standards.

8.9.2 Summary of Gaps

- Support of accounting
- Deployment of some appropriate amalgam of GridWeaver and LCFG for e-Science wide fabric management
- Integration of cluster (Grid fabric) management tools like LCFG and MDS style (section 7.6.2) registration
- Development of a Gridmake linked perhaps to LCFG

- GridPP has developed some useful tools GridSite and SlashGrid which could be broadly deployed (appendix section A.2.9.1.5)
- Mobile code brings non trivial security and correctness issues where there has been some initial considerations discussed in the appendix, section A.2.9.2.2(b).
- Broad based visualization services that scale to support large repositories and results of HPCC simulations
- Computational steering has been successfully emphasized by RealityGrid but this needs continued development ([RealityGrid] and appendix, sections A.2.9.1.3, A.2.9.2.6(b) and A.2.9.2.2(c)).
- Both visualization and code/data coupling projects will need standards for representing irregular meshes in same way NetCDF [NetCDF] and HDF [HDF] support regular meshes (this is related to workflow)
- The UK should continue work in Grid economies (see section 8.9.3.1 below)
- Improve collaboration (Access Grid [AccesGrid]) technology

8.9.3 Selected Snapshots of Current Activities

8.9.3.1 Grid Economies and Markets

To date, much of the effort in Grid Computing has focussed on making resources (e.g. supercomputer cycles, databases) available over the Grid, and also upon the discovery of those resources. Despite the clear shared understanding that Grid computing will not be free, the areas of accounting, tracking and charging for Grid resources have received little attention in comparison. Yet, until there are standard mechanisms for charging for resource usage on the Grid, the vision of users transparently accessing a world of resources about which they previously knew nothing of, will remain a dream. Resource sharing will never be able to move beyond the distributed project, or Virtual Organisation boundaries.

The problem is most acute in the UK, and more generally Europe, where the use of resources such as supercomputers has to be recorded and tracked in minute detail. At CSAR, Manchester's national supercomputing service [manCSAR], resources are allocated and charged for in "CSAR tokens" which have a notional real money cost – as an example, 1 token equates to less than 20 CPU hours on the 512 PE Origin 3800. While it would be desirable to open these machines up to increasing amounts of Grid usage, this is not possible until such usage can be automatically accounted, processed and charged for. Grid access remains limited to those who either already have accounts, or to whatever the service can afford to give away in a sort of goodwill gesture to the community.

This is unfortunate considering the fact that the service encounters peaks and troughs in usage. It would be valuable for us to be able to sell off excess cycles (which are otherwise lost). Similarly, during peak times it would be valuable for our users to be able trade their cycles in for cycles at other, quieter centres. Thanks to human diversity, peaks and troughs around the world tend to happen at different times; in a world-wide Grid-enabled future, we might even be able to keep utilisation high over Christmas and New Year.

The UK Market for Computational Services project [UKeSMarket] will attempt to address this gap, implementing mechanisms for producing and accessing resource usage information, and for making chargeable Grid services. It has two main activities: the development of tools and services to support the trading of Grid Services using the Open Grid Services Architecture (OGSA), and the exploration of the business models enabled by such an infrastructure. The project will build upon the Open Grid Services Architecture [OGSA], and will allow any OGSA service to be charged for via a wrapper mechanism, without requiring the modification of the service being charged for. This will also enable services to be bundled and resold by third parties, while still respecting the policy of the resource provider. (It is worth noting that the introduction of Reselling into Grid Computing is of great interest to the commercial sector.) The project is strongly linked to the standards being developed in relevant Global Grid Forum Working Groups, i.e. the Usage Record, Resource Usage Service (RUS-WG) [GGFRUS] and Grid Economic Services Architecture Working Groups (GESA-WG) [GGFGESA], as well as the Open Grid Services Architecture Working Group itself. Alongside this standardisation activity, the project will develop a reference implementation of these services and deploy them over the UK e-Science Grid. Its ultimate goal is to develop an infrastructure that will allow users to seek out services, which are provided for profit by

organisations in the Grid, evaluate the quality of the service and its cost, before using the service that best satisfies their specific requirements.

Further details can be found in the appendix, sections A.2.9.1.1 and A.2.9.1.6.

8.9.3.2 Improve collaboration (Access Grid) technology

The Access Grid [AccessGrid] comprises resources that support human collaboration across the Grid. In particular the infrastructure can support large-scale distributed meetings and training. The resources available include multimedia display and interaction, notably through room-based videoconferencing (group-to-group), and interfaces to grid middleware and visualisation environments. Applications may be shared so that the same information is available to all participants. The nodes themselves are dedicated facilities that support the high quality audio and video necessary to provide an effective user experience.

Access Grid meetings support information sharing and exchange. Also events in one space can be communicated to other spaces to facilitate the meeting, and they can be stored for later use. At the simplest level, this might be slide transitions or remote camera control. New forms of information may need to be exchanged to handle the large scale of Access Grid meetings, such as speaker queues, distributed polling and voting.

The CoAKTinG project ('Collaborative Advanced Knowledge Technologies on the Grid' [CoAKTinG]) is looking to improve the collaborative aspects of Access Grid technology. It aims to do this by integrating intelligent meeting spaces, ontologically annotated media streams from online meetings, decision rationale and group memory capture, meeting facilitation, issue handling, planning and coordination support, constraint satisfaction, and instant messaging/presence [Shum02A]. It should be noted that CoAKTinG requires ontologies for the application domain, for the organisational context, for the meeting infrastructure and for devices that are capturing metadata.

The original version of this contribution can be found in the appendix, section A.2.9.1.2.

8.9.4 Possible Future Projects

8.9.4.1 GridWeaver: an enhanced LCFG for e-Science wide fabric management

A current e-Science activity involving HP and Edinburgh is developing enhancements to LCFG [LCFG] which is a key part of future Grid technology. GridWeaver [GridWeaver] uses SmartFrog (Smart Framework for Object Groups) from HP. SmartFrog is a framework for describing the configuration of a service and deploying and managing the service through its lifecycle [SmartFrog]. SmartFrog consists of a language system for describing service configurations, a runtime deployment system and a component model to control deployed components. GridWeaver addresses:

- *Scale:* > tens of thousands of systems participating in each fabric
- *Diversity:* highly heterogeneous hardware and software environments; volatile and non-volatile network connections; dedicated and non-dedicated hardware, …
- *Complexity:* highly complex configuration of individual fabric elements, and of distributed software services executing across multiple fabric elements
- *Dynamism:* the fabric will change continually due to changes in hardware, software and due to failures
- *Validation:* validation is needed to ensure correctly configured fabrics, at all levels from individual hardware elements up to distributed services
- *Security:* Security moves up the priority list, especially for commercial workloads

Further details can be found in the appendix, section A.2.9.2.1 and A.2.9.2.2(a).

8.9.4.2 Development of a Gridmake

Thinking about building large C-style programs, there are two clear capabilities needed by *Gridmake*:

a) Making sure everyone "making" stuff is doing so correctly
b) Deciding who should make what without lots of communication

For example in Sun's JVM source tree, (a) is dealt with by shipping most of the tools with the source tree, so that it only relies on things that can always be assumed safe on the build machine. e.g. it picks up 'cc' (checking it's version x.y.z of the Sun compiler) but a 'bootstrap' JVM used in the build is shipped with the source tree. Similarly things like Linux, Nemesis, Xen are usually built taking all of the header files from versions shipped with the source rather than picking up whatever's in /usr/include. The usual approach therefore seems to be a kind of manual 'pack and go' system.

For (b) we imagine some kind of distributed work sharing queue would be reasonable. Machines building stuff take work off the shared queue, build it and possibly put new work back onto the queue (e.g. if they've just built something that other stuff depended on). Maybe some kind of feedback based system to decide what granularity of work unit is reasonable; batches of files rather than individual files?

The original version of this contribution can be found in the appendix, section A.2.9.2.3.

8.9.4.3 Debugging tools

"Grid-Debug": A project or program of work in this area would involve investigating techniques for software debugging. There are two areas, for example, that Cambridge University would propose looking at, which have hitherto received little attention from the Computer Science community. The first area is controlling complex multi-process applications through a single cohesive debugging interface. We can do this by virtualizing the resources used by the system, thereby allowing the threads that it involves and the network links that it uses to be modelled as a single controllable entity. This method will be applicable for moderately sized systems of perhaps a dozen nodes.

The original version of this contribution can be found in the appendix, section A.2.9.2.4, which also discusses a "Grid Log".

8.9.4.4 Grid-Based Visualisation Services at Leeds

Work at Leeds over the past few years has sought to extend the traditional modular dataflow visualization systems (such as IRIS Explorer, AVS and IBM Open Visualization Data Explorer) to meet the challenging requirements of modern computational science. A major driver from Grid computing is to distribute the processing pipeline, in a secure manner, between the desktop display machine and remote Grid resources. This has been realised as an extension of IRIS Explorer, in which the construction of the pipeline is carried out at the desktop, but some modules (such as simulation code) execute remotely on a Grid resource while others (such as the renderer and other 'user-facing' modules) execute on the desktop. Globus provides the middleware to support this distributed execution. A demonstrator for the e-Science program has illustrated how computational steering can be achieved in this way. This work has been done within the e-Science gViz project (involving Leeds, Oxford, Oxford Brookes, CLRC, NAG, IBM and Streamline Computing) [gViz].

A major visualization requirement from e-Science is to allow teams of researchers to work together, over a network, in visual analysis of their data. The dataflow paradigm extends naturally to allow each team member to execute their own pipeline, but to share data with collaborators – this data being exchanged at any point in the pipeline, as best suits the needs of the collaboration or the bandwidth of the network. Again this has been realised in terms of IRIS Explorer, as the COVISA toolkit [COVISA] [COVISAG]. Note however it is limited to synchronous collaboration, and both parties must be running the same visualization system.

More generally, in order to allow better collaboration between scientists, we need to define standards for the exchange of visualization data, and the exchange of visualization programs. Within gViz, we are starting to explore the use of XML to describe datatypes, and dataflow networks, work that may eventually lead to international standards for visualization.

Another important gap in current provision is support for maintaining an audit trail of a visualization session, so that the process by which a visualization was created may be recorded – and is thus repeatable. This notion of history would also enable us to provide support for asynchronous collaborative visualization, that is collaboration over a period of time, when one team member could take over from another at a later stage.

Another gap is the co-scheduling of resources for interactive large-scale visualization – where we need simultaneously access both to high performance visualization (through a VR facility) and also to high performance computing – a problem that becomes even more demanding in a collaborative setting.

All visualization systems are challenged by very large datasets. There is a need to bring together expertise in data mining, and expertise in visualization, so that the best combination of machine and human visual processing is used to extract knowledge from these large datasets. Specifically for visualization, research has begun – but more is needed – on algorithmic advances that increase the speed of conventional techniques such as isosurfacing and volume rendering – either by using parallelism or by intelligent filtering of the data.

The original version of this material can be found in the appendix section A.2.9.2.5.

8.9.4.5 Visualization and Computational Steering at Cardiff

Some groups including Leeds, CLRC, and Stuttgart are attempting to embed grid enabled applications within the Access Grid environment [AccessGrid]. This can be contrasted with the view that Access Grid type services (real time video, interactivity, CSCW or Computer supported Cooperative work) should be embedded in the generic Grid and the existence of these two separate systems/services should be removed over time.

Critical elements are the co-allocation of resources (scheduling of services) and network bandwidth aware applications. Other issues centre on security and the Grid support of "group".

The original version of this material can be found in the appendix section A.2.9.2.6.

Visualization:
The use of visualization services to support supercomputing installations is now widely accepted and it is encouraging that recent reports on the future of HPC provision in the UK have highlighted these needs. Traditionally the graphics supercomputer (shared memory architecture) has been tightly coupled to the numerical supercomputer. With the advent of the grid, and indeed to use of large clusters for numerical computation, the amount of data is unlikely to be able to be funnelled into a single shared memory graphics subsystem. There is a need, therefore, to investigate distributed graphics/cluster graphics architectures which perform the rendering operations and transmit to the user appropriately transformed data. This could be in the form of images (such as SGI's VizServer [VizServer]) which have the advantage of being independent of the generating database size and have a fixed and predictable format allowing bandwidth and Quality of Service issues to be addressed. Some of these issues are being addressed as part of the RAVE e-Science project at Cardiff University.

There is a need to:

- define and construct a visualization ontology.
- promote visualization issues within the Global Grid Forum.
- construct middleware which supports the composition of visualization services.

The above were outcomes of NeSC meeting on Grid and Visualization held January 23 2003 [UKeSViz].

Computational Steering
Here this term means "steering through the image" i.e. the user should be able to manipulate some element of the visualization directly to change a parameter within the numerical simulation.

Such systems require coallocation of resources. There are essentially two feedback loops in such systems. Visualization and the simulation update path. Ideally both should operate in real time. In this case we have a true "Virtual Environment" architecture. However, unless the fidelity of the simulation is reduced the second feedback loop is often updated at a much slower rate. Users are very tolerant of slower updates in this pathway provided they can explore the datasets rendered interactively, i.e. the visualization pathway has to be operated at interactive rates.

With respect to the visualization pathway, the same issues regarding the funnelling of data/graphics as indicated above applies - the real time elements just further stress the system. With respect to the simulation update pathway, we must also be able to predict (and impose) an update period of the simulation loop to ensure results are returned within an acceptable time delay. Resources must be co-allocated in terms of visualization elements and computational elements which gives rise to HCI and collaboration issues.

A prototype "Interactive" grid needs to be established to allow the community to develop and test ideas in these spaces.

A remaining issue centres on access to significant computational resources to allow high fidelity (interactive or at least reactive) simulations. Ideally this would just be a "reactive/interactive grid service" and the issue of which machine (or set of machines) to run the code on would be removed from the user.

Use of National HPC resources
At present to get high fidelity simulation for computational steered applications really requires the use of national facilities. However, it is difficult to gain "interactive" time on national facilities. There is a need to establish sufficient computational capacity in the Grid supported by aggressive interactive scheduling policies which can checkpoint and suspend batch jobs to give priority to interactive users.

There is a need to investigate the "efficiency" of national computational resources. The batch mode may keep the machine busy and achieve high machine utilization rates. However, if the run result is discarded by the scientist for whatever reason (failure to converge, parameter errors etc) this is not "useful" work. A computational steered application may however "idle" machine resources and appear inefficient in terms of machine utilization but may avoid expensive wasted runs of inappropriate parameter space due to the interactive nature of the process. A study to compare and contrast the batch mode with computational steered applications in the support of scientific understanding and processes needs to be conducted and this will require resources to be allocated on the above basis. This would truly reflect one of the aims of the e-Science programme in exploring and supporting "new ways of doing science".

8.9.4.6 Grid Visualisation Services at CCLRC

The Grid Visualisation Group within the CCLRC e-Science Centre is developing application visualisation services based on a Grid/Web services model. The suite of software modules and associated application programming interfaces to access them are packaged as the GAPtk (Grid Applications Portals toolkit [GAPtk]).

The original version of this material can be found in the appendix section A.2.9.2.7.

8.9.4.6.1 GAPtk visualisation server

At the heart of the GAPtk toolkit [GAPtk] is a central application visualisation server, which mediates between a variety of desktop applications (custom software), problem solving environments (e.g. MATLAB) and portals (Browser based interfaces) and a variety of third party data access services and the Grid fabric layer. It adds valuable intelligence to the communication layer and hides the complexities of distributed computing. The server also contains some generic application and visualisation services, for instance, server-side rendering of publication quality images or generating an animation sequence from the geometry computed. The server also contains software modules that coordinate the input and output from various external services, adding context sensitive semantic information to the outputs before sending these to the clients.

The GAPtk server has three interfaces. The server's client interface uses a Web services communication protocol, except for very large datasets. The server's data interface to third-party data query and search Web services again uses Web services communication mechanisms. A third server interface communicates with the Grid fabric layer to obtain secure data and compute resources using appropriate Grid and Web protocols such as GridFTP and SOAP. A diagram showing the overall architecture of the toolkit can be found at http://ws1.esc.rl.ac.uk/images/projects/gaptk/GridVisDiagram2.pdf.

Generic application and visualisation services are implemented as atomic components within the server with the objective that an application developer should be able to compose a complex visualisation scene from these modules. We see this as an efficient way to implement flexible higher order data visualisations as the user will be able to compose and superpose the variables of his problem space within a visualisation scenario. As part of this, we support the variability and scaling required in each data dimension for each variable as appropriate for a given application context to make the complex visualisation scenario semantically correct, intuitive and easy to explore. Towards this end, we are developing a visualisation metadata schema to record data characteristics that are essential to map the visualisation geometry onto a rendering pipeline. This allows an application programmer to create geometry with superposed data structures that will render these with appropriate scaling and behaviour at the client end. We are also working on Grid-enabling basic visualisation algorithms to support near real-time data exploration.

8.9.4.6.2 Application programming interfaces

Three separate high level scripting interfaces for application programming are being developed for the services and interfaces of the GAPtk server described above.

On the client-side desktop, it is possible to construct customised task-based user interfaces for a wide range of applications using a variety of problem solving environments (PSE) with these high level scripting interfaces. Our goal is to help users build on existing knowledge and investment in application specific problem solving environments as well as to support users who wish to develop their own more advanced Grid-aware environments, for instance for distributed, collaborative computational or experimental steering.

Our services and software are generic and wherever possible a thin client interface layer is provided to easily link into these services. This provides the flexibility to use a visualisation toolkit such as IRIS Explorer as a component of a collaborative problem solving environment as an alternative to coupling an application and its collaboration requirements within the visualisation toolkit. This allows each partner in a collaborative session to connect their own user interfaces, problem solving environments and portals to the server services and share their applications and interaction.

The other two server-side APIs allow new Web services, applications and Grid fabric layers to be linked in. Where an application is already either Grid-enabled or parallelised, a simple services based wrapper enables it to be linked into and made available via the GAPtk framework.

8.9.4.6.3 Applications

The Grid Visualisation Group is currently building two applications of GAPtk, one for environmental sciences, in particular for oceanographic diagnostics visualisation incorporating data assimilation and the other in condensed matter physics, where the analysis of experimental data feeds into simulation whose output helps to decide further experimental settings.

8.10 Portals and Problem Solving Environments

8.10.1 Introduction

A Grid Computing Environment (GCE) has been defined as "a set of tools and technologies that allow users easy access to Grid resources and applications" [Fox03A]. A Problem-Solving Environment (PSE) is an integrated software environment for the computational solution of a particular problem, or class of related problems. Thus, whereas a GCE is quite generic, a PSE is specific to a particular application domain – one might have different PSEs for, say, molecular dynamics, financial modelling, battlefield simulations, and so on. A goal of a PSE is to provide high-quality, reliable problem-solving power to the end user without the need for them to pay attention to details of the hardware and software environment not immediately relevant to the problem to be solved. Many recently developed PSEs make use of Grid resources and are closely related to the GCE concept. These types of PSEs can be viewed as GCEs with add-ons specific to some particular application domain. These add-ons might include domain-specific knowledge and information resources, ontologies, tools, and data repositories. An "intelligent" PSE may have many of the features of an expert system and assist the user in formulating complex problems, running them using the appropriate resources, and visualizing and analyzing the results. PSEs are often used collaboratively to solve multi-disciplinary problems. The term "application portal" is used as a synonym for PSE.

PSEs are used in areas such as design optimization, parameter space studies, rapid prototyping, decision support, and industrial process control. However, an important initial motivation for their development was their support for large-scale collaborative simulation and modeling in science and engineering, for which the ability to use and manage heterogeneous distributed high performance computing resources is a key requirement. A second important requirement of such a PSE is that it should provide to the end user easy-to-use problem solving power based on state-of-the-art algorithms, tools, and software infrastructure. These types of PSE can reduce software development costs, improve software quality, and lead to greater research productivity. These effects in turn result in better science, and in the commercial sector better, cheaper products that are brought more rapidly to market. PSEs have the potential for profoundly changing the way distributed computing resources are used to solve problems – in the future it is expected that web-accessible PSEs will become the primary gateway through which distributed computing resources are used. We noted in section 7, that PSE's use many Grid technologies in other of our 11 technology categories in sections 7 and 8. In particular workflow in sections 7.4 and 8.4 is critical in most PSEs as the glue (software bus) for the different parts of the system.

A user portal provides an interface to GCE components that front-end lower-level Grid resources. This can be viewed as a sort of shell for accessing GCE tools and services. Such a "GCE Shell" [Fox03D] would support running and compiling jobs, moving among file systems, and so on. The Unix shell model is also used in Legion [Natrajan02A] and JXTA [JXTA] to create Grid capabilities. Unicore [Unicore-A], the NPACI Grid Portal Toolkit, HotPage [Thomas03A] all provide examples of GCE Shell portals. Most PSEs would typically have a GCE Shell built into them.

The Globus Toolkit provides little direct support for building GCEs. However, Commodity Grid (CoG) wrappers [Laszewski03A] and the Grid Portal Development Toolkit (GPDT) [Novotny03A] can be used as building blocks of full GCEs that use the Globus environment for basic Grid services. At the last Global Grid Forum GGF7, the GCE research group held a workshop discussing component models for portals stressing recent experience using JetSpeed [Jetspeed] and GridSphere [GridSphere] from the GridLab EU project [GridLab]. This GGF process is documented in terms of GGF information documents [Fox03B] which are being extended based on the GGF7 workshop [GGFGCE-C] and adoption of this type of approach should lead to greater re-use of portal components between e-Science projects.

8.10.2 Summary of Gaps

- Need general framework/design patterns (and consensus as to their adequacy) for problem solving environments and portals
    - This could be based on Information documents produced by the GCE Research Group of the Global Grid Forum

- Identify/refine toolkits for building portals and PSE's in a much quicker fashion for new applications
    - This will be critical in making Grids/e-Science attractive to broad user community
    - Technology like Jetspeed (recommended by GCE research group of GGF) not well known in UK program.

8.11 Grid-Network Interface

8.11.1 Introduction

The Grid puts special requirements on the Network because often the Grid requires predictable Quality of Service (QoS) to ensure that services can be coordinated in an end-to-end (or more precisely service-to-service) fashion. This has implications for both the Grid runtime and its integration with the network layer. If the Grid has as discussed in section 6.2, a clearly defined messaging substrate then it is relatively clear how one can integrate network and Grid runtimes. The capabilities needed were pioneered by the well known Network Weather Service [NWS] and fall into two areas. One needs to be able to monitor and predict network performance; secondly one has to be able to use this performance information to either reserve network capability or if this is not supported, one needs to estimate which routing strategy will most reliably realize ones goals. This must be achieved in the typical dynamic multi-supplier multi-technology network and take account of performance degradations and security issues at firewalls. It is likely that in general different protocols could be used in different parts of the routing and suggests that decisions as to routing and protocols should be made at the messaging and not the service layer. The latter should have API's to specify priorities and expected bandwidth/latency requirements.

Networks have developed the concept of NOCs (Network Operation Centers) to manage the operation of networks coping with hardware issues and security problems like denial of service attacks. It is possible that Grid Operations Centers (GOCs) will emerge adding control of the core Grid resources (servers and message brokers) and at least this concept should be explored. This is related to the requirement of service-to-service QoS which is challenging when the country backbone and regional networks are under different control.

More details on the interaction of Grids and networks can be found in the appendix section A.2.11.1.

8.11.2 Summary of Gaps

- Holistic and flexible Performance monitoring and diagnostic Architecture. Detailed capabilities of this network infrastructure include
    - A set of performance measurement points (PMPs) not just at the edge, but throughout the Grid fabric
    - A well defined and common ontology for network measurements (and perhaps their schemas)
    - Ability to store and replay complete traces
    - Well defined "service" fronts to these PMPS (possible Grid services) which give out measurements against proper credentials
    - A set of clients which give different views onto this information for different users
    - This effort includes improved Network Weather Service
- See also comments on messaging and transport infrastructure
- Need to treat network as a resource and enable network resource reservation and claiming.
    - This is related to Grid Workload management and Grid data management (replication optimization).
    - There are important authorization issues
    - Need to develop middleware API's to network quality of service

8.11.3 Selected Snapshots of Current Activities in Monitoring and management

8.11.3.1 Network Measurement and Monitoring

As the Internet grows larger and heterogeneous, the ability to measure the real-time properties and service quality experienced by the user traffic becomes of critical importance. Measurements reflecting the actual service experienced by the user traffic can prove valuable for long and short term network design decisions, traffic engineering, as well as for Service Level Agreement (SLA) negotiation and policing. Current measurement infrastructures focus either in measuring the path properties of special-purpose traffic (active measurements) or in correlating and analyzing one-point, link traffic (passive measurements). Acknowledging the need for a more service-centric measurement instrumentation that would assess and categorize the properties experienced by the different user applications and protocols, a possible way ahead is the development of two-point, in-line measurements - a novel measurement infrastructure that can operate end-to-end and reveal the effects of the network behaviour on the user traffic, for the Next Generation (NG), IPv6 Internet. Avoiding artificial measurements and reducing the consumption of network and computational resources from the collection and transfer of measurement data itself are some of the major challenges that these in-line techniques address. Today, the requirement for migrating to IPv6 as a universal transport mechanism has finally been established, mainly due to the fast consumption of the IP address space, influenced by the growth in new markets requiring IP connectivity (particularly mobile and wireless). These in-line techniques focus on all-IPv6 networks and perform measurements at the ubiquitous IP network layer, by exploiting the concept of the IPv6 extension headers, allowing for more accurate, flexible and less intrusive service measurements to be carried out. The term 'inline' implies that measurement triggers, which invoke measurement activity and the measurement data itself, are piggybacked onto real user packets, virtually guaranteeing that they experience the same treatment with the actual traffic.

Measurement and monitoring infrastructures fall into two main categories, namely active and passive. An active measurement is one in which part of the measurement process is to generate new traffic and inject it into the network, while a passive measurement is one in which existing network traffic is recorded and analyzed [Paxson96A]. Active techniques rely on injecting traffic with known characteristics into the network to test particular attributes of a service. This type of measurement technology is employed to make 2-point measurements, particularly in relation to response time, loss, bandwidth and availability. Active measurements architectures are mostly being based on variations of the ping and traceroute programs. They try to measure the performance of wide-area network connectivity among the participants [Matthews00A] [SurveyorNetwork], by sending Internet Control Message Protocol (ICMP) packets [IETFICMP], and implementing metrics defined within Internet Engineering Task Force (IETF)'s IP Performance Metrics (IPPM) Working Group [IPPM]. Variable and fixed-size UDP packets are also being used to measure properties such as one-way delay and loss [RIPENCC] [Georgatos01A]. There also are dedicated protocols, like NLANR's IPMP (IP Measurement Protocol [IPMP]) for performing active measurements, designed to overcome the weaknesses of existing protocols (e.g. ICMP), currently used by other infrastructures [AMPNLANR], but unfortunately these risk receiving preferential treatment by the network. The major limitation imposed by active techniques is that they measure the forwarding and routing behavior experienced by synthetic injected traffic and not necessarily by the real user traffic. It is well-known that within a network node, ICMP can be given higher or lower priority to the rest of the traffic, and may therefore make the network appear to have either poorer or better performance than it actually has. Likewise, using UDP packets for synthetic traffic can also lead to inconsistencies between the observed and the actual performance, mainly due to the bandwidth starvation caused to TCP flows by the "unresponsive" UDP bursts [Floyd99A], or due to specific policies possibly applied to UDP traffic. Furthermore, the periodic sampling used by some architectures has the limitation of not observing periodic network events that occur at times other than when the samples are scheduled [Matthews00A]. Additionally, the extra network load produced by the generation of the special-purpose traffic might itself be a factor in measuring a poorer performance than the network actually delivers. Passive measurement technologies observe real traffic on a link without disruption to the service. Typically products that work at this level run state machines, which perform some initial level of filtering to trace the traffic of interest and then search for particular events using pattern-matching techniques [Hegering99A]. Tools also exist to extract payload data from packets that match the specified pattern [CiscoIOS]. The collected data is then made available to

interested third parties through either a pull (SNMP, CMIP based products) or a push model. As with active measurements, the concern with both pull and push based models is that users may end up generating immense amounts of measurement data that need to be shipped across the same links being monitored. Full packet capture is also possible although this may itself degrade the performance of the service under test.

Other passive monitoring techniques, which rely on events generated from a particular traffic stream, are better suited to address Quality of Service (QoS) performance related issues [Jain86A] [Claffy95A] [Brownlee97A].. Passive techniques are particularly useful in gathering 1-point measurements of real user traffic at an observation point on the network. However, they are less suitable for making 2-point measurements of real user traffic, such as one-way delay, due to the complexity involved in correlating the samples collected at two distinct observation points. Techniques are also required to ensure that both observation points trigger on the same packet for the collection of measurement data. Error handling for lost or mismatched samples is also necessary. Furthermore, passive measurement probes may not be able to keep pace as traffic volumes increase and data rates get faster. Passive measurements are considered to be highly accurate and usually avoid artificial results, since they operate on the actual traffic of the network. However, for some metrics it is exceedingly difficult to see how one can measure them passively (e.g. route taken by packets) [Paxson96A] and scalability arises as a significant restraining factor.

8.11.3.2 Network Information Services

As discussed in Sec. 7.11, the current situation is that we are at the very beginning of seeing the availability of novel services technically on the wide area network, and we are similarly at the beginning of seeing some of the control plane software necessary to give access to grid middleware.

A (non-exhaustive) list of clients for network information includes Grid users (to determine when and where problems exist prior to reporting to a support or operations centre), Grid middleware (e.g. resource brokers, data management) and Grid operations centres and network operations centres (to monitor performance and service level, and to diagnose problems). A grand vision to cater for all of these (and future as yet unknown clients) is to implement a network performance and characteristic measurement architecture based upon well defined services and interfaces at different levels such that progressively more complex services can be built up from components. In this way the end-user's viewpoint and the network provider's viewpoint could be combined in a "holistic" way. Thus future work in this area should allow:

- Use of a heterogeneous set of monitoring infrastructures across different administrative domains whilst allowing each authority to have different outlooks and methodologies for the measurement of network characteristics.
- Use of this information through interfaces to be defined through an appropriate Grid service architecture such that measurements can be made available on the boundary of the domain of any authority by supporting some or all of the interfaces.
- Administrative domains to control which information they will give out against which authorization.
- Higher-level functionality to be built on top of lower-layer Network Grid services in order to satisfy the needs of the different clients listed above.

To give specific examples all of the pre-existing schemes of Sec. 7.11 could, with some work, be used as back-ends to low level network information services following the GGF working group in this area. These services could then be bound together to build a higher level "network information" service able to provide an end-to-end picture by combining information along the routes (this is more scalable than the current mesh structure) and publish this into Grid resource information services. The same low level services could also be built into a different higher level service for use by Grid operations centres in trouble ticket diagnosis.

8.11.3.3 Network Performance Services

The overriding rationale for the use of a hybrid of network services in the future is unassailable. The current "uniform best efforts IP only" service has worked so far due to over-provisioning and enforced limits upon high rate data transport coming from technical limitations. However we see that in the future the needs of a very large number of low demand users (type-A) needs to co-exist with a modest number of medium demand users (B) and a small number of very high demand users (C) who will soon be running Gbit/s scale flows. The key point is that type C, and some of type B, do not need layer-3 IP routing for much of their needs, thus to provide for all of these in a uniform way using layer-3 routed best efforts IP only would very likely be inefficient and unnecessarily expensive (layer-3 routing is an order of magnitude more expensive than layer-2 switching. This means a more appropriate network will combine a hybrid of IP routed-services (including QoS) for those who need them co-existing with layer 2 and below services where appropriate.

To make use of this model, the assignment of appropriate service levels must be done automatically by the interaction of Grid middleware with a network control plane, in accordance with policies agreed by the network's management. There will thus need to be a tight coupling between the Grid infrastructure and the network management infrastructure, in terms of access, control and measurement functions. The required services must be available end-to-end across multiple network domains and incorporate features such as advance reservation, authentication, authorization and accounting (AAA). Allocations must be restricted to authenticated users acting within authorized roles, the services available must be determined by policies derived from SLAs with user organisations, and the aggregate services made available to VOs must be monitored to ensure adherence, along with the performance delivered by the networks. Advance resource reservation and allocation is not restricted to networking and will be presented to the Grid user as a global resource service covering network, storage and computing. However, networking resource must be allocated end-to-end across multiple domains, which creates a complex problem of co-ordination and dynamic re-configuration of resources within a number of administrative domains. The control structures and network models implied by offering differentiated services with end-to-end resource allocation and advance reservation and allocation are not completely understood today. The granularity of resource reservations in terms of bandwidth and duration is important, together with required QoS (Quality of Service) parameters. Thus future work in this area should include:

- Implementation of a scalable AAA architecture acceptable to the network domains upon which the Grid runs.
- Proof of principle demonstrations of access to layer-3 diffserv-based services, extended layer-2 VLANs, based on gigabit-Ethernet (GE) connections and point-to-point switched layer-1 connections (STM-4 to STM-64, 1GE, 10GE) from Grid middleware layers.
- Similar access to secure channels (required by some applications).
- Scheduling (advance reservation) of all of the above
- Development of appropriate APIs within the GGF.

9 Education and Support Gaps

9.1 Gaps

- Lack of good documentation – particularly for Globus.
- Support of rapidly varying technology needs special attention which involves both universities as well as traditional support structure
    - Need a way to assess maturity and utility of new tools
    - Need to find out about the new tools!

- Develop stronger support center-user interactions about future developments and current Grid software releases
- Support center role and staffing needs to reflect broadening of Grid software base
- OGSA-DAI's great success implies need for long term support (as well as enhanced functionality listed separately)
- Grid Operations is not supported in the same network operations are
    - Need optimization and monitoring tools
    - Need ways to query running services to find operational state and other operational management services
    - Need a GT2/GT3 transition strategy with production Grids like White Rose particularly sensitive to this.
- Challenges from increasing breadth of software to be supported and change from supporting Grid developers to supporting users.
- Testbeds and well defined software engineering processes will be important
    - Emphasize that software produced in e-Science should have test procedures, sample applications and tutorials
    - Need to define UK e-Science deployment process
- Need production Grids and experimental testbeds

10 Research Gaps affecting near-term e-Science (issues where not enough is known to allow "development" project)

It is difficult to make a distinction between specific research gaps affecting e-Science in the near-term, and the types of technology gaps identified in section 8. Some of the research areas identified on section 8, such as GridMake (section 8.9.4.2) and GridDebug (section 8.9.4.3), could equally well have been placed in this section.

10.1 Gaps

- How does one support Personal Grids
    - Include personal Grid storage view in fashion familiar from NFS
- Understanding of differences between e-Business and e-Science requirements
    - Differences identified in security and workflow
- Need to develop experience and quantify best practice as to how to design services; this can be called "software engineering for service architectures"
- Service (system) management and administration (heart of autonomic computing) not addressed in current activities
    - Is enough known to make this a broad technology infrastructure gap or is further research needed?
- Dependability and fault-tolerance
- Understand relationship between provenance (history of the result of running a service) and "virtual data" (specify information in terms of workflow needed to create it). Develop workflow approach integrating ideas
- Debugging tools
- Optimization of Grid application performance
- One needs to develop the field of Grid benchmarking and performance measurement

11 Perception and Organizational Issues and Gaps

- When will GT3 be robust enough to deploy?
- Next set of projects must aim at decreasing time and expertise needed to deploy grids
- Rapidly changing field creates confusion as to what really is current technology state and what is expected from GGF EDG Globus IBM etc. in near future
    - Not clear what one should use and where a given group should invest effort

    - Not helped by reputation that Globus has for erratic release and software support
    - Could be addressed by some coordinated e-Science group supporting the dissemination of Grid/e-Science roadmap integrated over Global projects (see earlier comments under Education and Support gaps)
- More explicit interactions between different Data Grid projects could be valuable
- Some projects have very stringent application goals that run counter to producing software that has generic applicability even within a given application domain
    - Success is measured by demonstrations and not by production of generic software infrastructure
    - Next set of pilot projects should focus on generic software for appropriate application areas – not clear exactly how broad the application area should be
- Current e-Science portfolio has a lot of exploratory projects with informal inter-project coordination
    - Next round should be fewer larger projects emphasizing and explicitly funding inter-project coordination so as to produce generically useful software
- In spite of success of OGSA-DAI, there is a perception and perhaps reality that Grid effort is still compute and not data focused; particle physics has huge data needs but its requirements are different from areas like Bioinformatics which are more heterogeneous
- Risk from rapidly changing technologies
- e-Science timescale is very aggressive; for some projects at least too early to identify gaps reliably and one needs 2-3 years more experience
- Need a Grid model for software licenses
- GGF is slow and unclear route from discussions at GGF to implementations
- If a Service specification is simple (simplistic), then it is compatible with a more sophisticated service but there is danger that different groups will extend original specification in different incompatible ways. The simplicity of OGSI (for example in notification service) thus does not hinder Grid development but interoperability will only be at the simple level and so could be incomplete. This type of issue could get accentuated as OGSA starts defining functionality of different Grid Services
- OGSI defines the structure of Grid services (extending Web services to a stateful component) while OGSA will systematically define the functionality of Grid services. OGSI is essentially complete but OGSA is only just starting. OGSA-DAI is an important example of such a functionality definition.
    - This creates confusion as to significance of OGSA compliance in Grid Infrastructure and to projecting timescales for functional OGSA environments
    - Further confusion is engendered by the different Grid levels that can be separately implemented. One can use GT3 for example for its OGSI infrastructure but not adopt its particular OGSA services
- Some IT vendors consider aspects of current Grid technology such as OGSA as naively ignoring previous lessons from CORBA/DCE etc.
    - This emphasizes risk of getting trapped in technology backwaters not supported by commercial vendors
    - Note there are many commercial thrusts in distributed computing; Web services, .NET and Java for example. There will be good support in these areas
    - Currently the government is supporting Grid
- Understanding of interaction between GGF [GGF-A] and W3C/OASIS [W3C] [OASIS]and how this is synergistic as opposed to competitive
- Interaction between of Grid community with Digital Library, Agent and Semantic web communities/technologies
    - The Semantic web engagement has been excellently addressed by UK e-Science projects and the GGF Semantic Grid research group set up by de Roure and Goble. Need to monitor this to see if more work needed to take best advantage of Semantic Web technologies for the Grid
- Possible need for an ongoing "Gap analysis" to track progress in the "post GT2" world
    - Understanding of e-Science software approach which will lead to a broadly applicable e-Science and e-Business methodology allowing general deployment in 2006
    - Refine the different requirements of e-Science and e-Business

    - Continue the process to identify key Grid infrastructure and nurture (formalize interfaces, robustify software) it for this 2006 UK wide deployment
- Need to understand in detail the relative role of Industry and Government/Academia in developing and supporting the different aspects of Grid infrastructure
    - This affects both detailed OGSA components and optimization of core Grid infrastructure
- Could arrange the new initiative with four classes of activities
    - Application oriented activities producing generally usable software as opposed to that for a particular group
    - Specific "software nuggets" identified in the gap – such as improved OGSA-DAI, a particular workflow model
    - Core Grid software such as InterGrids, Security, invocation framework, messaging etc.
    - Grid support activities
- Grid federation technology enables individual projects to proceed and only be loosely synchronized with projects like GT3 as mediation technology can reconcile differences
- Who could staff new middleware initiatives?
    - Current activities consume a lot of available talent
- If you think just two or three projects will dominate in 2006 (say GT3, IBM and UNICORE), then it will not be difficult to "bid" for commercial support/replacement. Correspondingly if the 2006 scenario is more varied, there are difficulties in support unless as in InterGrids heterogeneity is explicitly addressed.

12 Acknowledgments

The authors gratefully acknowledge the following people who have contributed text for parts of Sections 3, 4, and 8.

Jim Austin	York University
Nicholas Avis	Cardiff University
David Berry	National e-Science Centre
David Boyd	Rutherford-Appleton Laboratory
Ken Brodlie	Leeds University
John Brooke	Manchester Computing
Howard Chivers	York University
Peter Clarke	University College London
Geoffrey Coulson	Lancaster University
Simon Cox	Southampton University
Jon Crowcroft	Cambridge University
Matthew Dovey	Oxford e-Science Centre
Alistair Dunlop	Capitol Radio Group
Yan Huang	Cardiff University
David Hutchison	Lancaster University
Mike Kirton	QinetiQ
Kerstin Kleese	Daresbury Laboratory
Bryan Lawrence	Rutherford-Appleton Laboratory
Tony Linde	Leicester University
John MacLaren	Manchester University
Luc Moreau	Southampton University
Steven Newhouse	Imperial College
Norman Paton	Manchester University
Omer Rana	Cardiff University
Lakshmi Sastry	Rutherford-Appleton Laboratory
Nigel Shadbolt	Southampton University
Ian Sommerville	Lancaster University
Mike Surridge	IT Innovation
Ian Taylor	Cardiff University
Paul Watson	Newcastle University

Part IV: UK Open Middleware Infrastructure Institute (OMII)

13 Core e-Science Middleware Action Plan

This action plan recommends the establishment of a UK Open Middleware Infrastructure Institute (OMII), as described below, to spearhead the development of a complete Grid middleware stack. It is expected that much of the software will build on existing prototypes for services documented in this report. The action plan will make use of agile software engineering practices as described in Section 13.2, and has a proposed project structure described in the start of Section 13.1. Note that we would expect this project to part of an international collaboration but the quality required in the software requires strong central leadership from the UK program.

13.1 Elements of the Action Plan

The proposed activities of the UK OMII are based on the assumption that grids will be constructed using a variety of service-oriented technologies and protocols. The UK OMII will ensure that these separate grid "islands" can interoperate. The UK OMII does not intend to develop a whole new system but rather evaluate existing Grid solutions and add software as needed to ensure a simple robust system supporting federation between different installations.

It is envisaged that the UK OMII will have two central components, together with associated satellite sub-projects. Thus, the main components are:

1) **Technologies Team.** This will maintain a repository of grid systems and a testbed for compliance testing. In addition, the technologies team will provide training for rapidly evolving technologies, and maintain a "technology watch" that provides proactive notification of new and expected changes in software concepts and technologies. The Technologies Team requires a staff of 6 people.
2) **Core Software Engineering/Coordination Team.** This will develop the Grid system architecture and core software, and use and support project-wide agile software engineering principles. It is expected that most of the work will be performed by distributed teams, with the Core Team providing the professional "glue". The Core Team will be responsible for identifying and exploiting the synergies between the different sub-projects. The Core Team requires a staff of 6 to 12 people and could be split between 2 sites.
3) **Satellite sub-projects.** These would be like the current UK e-Science projects with distributed teams but coordinated by the central components. Due to difficulties of combining academic responsibilities with project discipline in this or previous category, key university personnel may need to be "bought out" of their usual university teaching and administrative duties for the duration of the project.

The Project Advisory Board will meet frequently (once a month initially). The UK OMII will continue to participate in the Global Grid Forum and the definition of the emerging OGSA, as well as working with existing Grid projects, industry, the Globus Toolkit 3 developers, and the LHC Computing Grid project.

The different parts of the project will now be outlined. Note that one can identify a smaller set corresponding to a "minimal program" but this is not recorded here.

13.1.1 Grid Systems

The Grid Systems sub-project will continue participation in GGF emphasizing the requirements of e-Science middleware innovation, and work with existing Grid projects. A customizable Grid will be developed supporting an **Autonomic Grid** and one or both of the following:

1) **GridLite**: a lightweight Grid infrastructure that could be based on Jini.
2) **P2PGrid** : a peer-to-peer Grid infrastructure that could be based on JXTA. P2PGrid could also fulfil the requirements of GridLite.

13.1.2 Basic Technology

The Basic Technology sub-project will investigate the federation of Grids using best practice in modern middleware development. Mediation between the Grids will be based on OGSA. The following areas will be looked at:

1) Asynchronous messaging with metaobject protocol (MOP) support of message wrappers will be used for security, management, and virtualization of binding, routing, and destination. Support synchronous collaboration will be included. Scalable fault-tolerant management, including notification and provenance, will be examined, as well as network monitoring and scheduling.
2) The federated security infrastructure, with good certificate support and fine-grain authorization, will be an important aspect of this sub-project.
3) The execution environment component of this sub-project will further develop the concept of an "e-Science Bean" in which the concept of a Enterprise Java Bean will be adapted for use in e-Science. In addition, invocation frameworks will be investigated.

13.1.3 Essential Services

The Essential Services sub-project will investigate:

1) Workflow runtime, supporting both transactional and dataflow styles of service interaction. This work will use input from existing activities, including OGSA-DAI, and datamining and visualisation application areas.
2) A Federated Distributed Information (FDI) system supporting service registration, security meta-data, replication and other file meta-data, accounting meta-data, and generic service meta-data. A fabric management interface also will be developed. The requirements of the Semantic Grid and related provenance issues will be examined. Both Grid (based on OGSI and Service Data Elements) and regular Web services will be supported by the FDI. Should support 3 mechanisms – central catalogue as in MDS or UDDI; P2P style messaging; and distributed service look up where metadata stored in individual services as SDE's.

13.1.4 Core Domain Grid Services

This sub-project aims to meet the requirements of Campus Grids and Particle Physics (LCG/EDG) Grids, as well as the Hybrid Grids needed in Astronomy, Bioinformatics, and Environmental and Earth Science, and will investigate the following areas:

1) An Information Grid based on OGSA-DAI and including support for sensor and other realtime data streams, as well as lightweight and heavyweight transformation services based on "Java" filters and the Compute/File Grids respectively.
2) A Compute/File Grid that incorporates a meta-scheduler and management infrastructure supporting Condor, and optionally the Sun Grid Engine. The Compute/File Grid should also provide an interface to remote files and tapes. Replication (caching) should be supported.

13.1.5 Programming Environments

This sub-project will examine component models and workflow tools and support for e-Science with the aim of developing a Scientific Workflow Execution Language (SPEL) to contrast with BPEL aimed at business workflow.

13.1.6 Portals and User Interfaces

This sub-project will develop component-based user interfaces based on JetSpeed and "GridSphere" portal aggregation services. GridShell and problem-solving environment tools will be supported.

13.1.7 Other Grid Services

Several higher level Grid services can be built on the middleware framework described in the sections above. Important projects include Semantic Grid, .NET-based Grid, Computational Steering, different Programming models, Provenance Tools, Collaboration, Debugging, Gridmake and the GriPhyN Virtual Data). A peer-to-peer Grid could be placed as an independent project if not part of the central activity.

13.2 Agile Development of Grid Middleware

The goal of agile software engineering is to apply agile development techniques – particularly test-driven design, refactoring, pair programming, and short, fast design iterations – to the production of high-integrity Grid middleware. Agile development techniques are essential in addressing the specific concerns of Grid middleware, particularly time-to-release, dealing with legacy systems and application integration, and the need for frequent refactoring because of changing toolkits and underlying infrastructure. Agile techniques may be particularly applicable for building high-integrity Grid middleware, because of their emphasis on test- and analysis-driven development. Critical issues include:

- Establishing detailed, precise requirements for high-integrity Grid middleware.
- Determining appropriate validation and verification technologies to apply in order to provide assurance to users of the integrity of the middleware. Technologies from a continuum of lightweight and automated testing and analysis approaches, through to heavyweight model checking and theorem proving, may be considered.
- Specifying and implementing an agile development process for producing, verifying, and validating Grid middleware.

Part V: Appendices

14 Detailed Descriptions of UK Grid Services and Activities

This is available as a separate document called Appendix A. Subsection x.y of this is referenced as A.x.y and not 14.x.y in the other sections.

15 Worldwide Grid Resources

Many of the brief overviews given below in sections 15.2 to 15.4 are taken (often verbatim) from the web site following each overview, and these sources are gratefully acknowledged.

15.1 Glossary of Concepts

API. An API (Application Program Interface) defines a standard interface (e.g., a set of subroutine calls, or objects and method invocations in the case of an object-oriented API) for invoking a specified set of functionality.

Autonomic Computing. An autonomic computing system is one that is resilient, and able to take both pre-emptive and post facto measures to ensure a high quality of service with a minimum of human intervention. An autonomic system has some ability to self-diagnose and self-repair system faults and anomalies.

Dynamic reconfiguration. The ability to replace components of, or add components to, an executing system without stopping the system execution.

e-Science. e-Science refers to science that is enabled by the routine use of distributed computing resources by end-user scientists. e-Science is often most effective when the resources are accessed transparently and pervasively, and when the underlying infrastructure is resilient and capable of providing a high quality of service. Thus, *grid computing* and *autonomic computing* can serve as a good basis for e-science. e-Science often involves collaboration between scientists who share resources, and thus act as a *virtual organization*.

Federated service. A service is federated if multiple distributed instantiations of it appear to other services as a single integrated service with either the same or different interface. Often one terms this a "meta-service". For example, federated databases and meta-schedulers are relatively familiar concepts. Only services with broad scope need to be federated.

Federation of Grids. A set of Grids are considered to be federated if they consist of *interoperable services* that are federated when a service needs to cross Grid boundaries.

Grid Computing. The Grid embodies a vision of a global computing and communications infrastructure providing transparent and pervasive access to distributed resources, such as compute cycles, storage, data/information repositories, instruments, sensors, and human expertise. Transparency implies that users are unaware of the location of the resources they use, and in some cases of what specific resources they use. Transparency supports the illusion that the distributed resources reside in the user's local environment. Pervasiveness implies that the resources are accessible through a wide variety of network-enabled devices.

Grid Economy. A Grid Economy is an environment in which resources and services can be freely charged and paid for. In a Grid Economy it should be possible for resources and services to be bundled and resold by third parties (where permitted by the resource provider). All of these activities must respect the policies of resource provider, end-user, and supply-chain members.

Grid Service. The Grid service concept extends Web services with additional capabilities and interfaces, such as an information port and a notification port.

Idempotent interface. An interface is said to be idempotent if repeated invocations after the first have no further effect.

Interoperable Grid. An interoperable Grid is a Grid built from services that communicate between ports using interoperable standards.

Interoperable service. An interoperable service is a generic service supporting some agreed interfaces that allow this service to be provided by any standards-compliant build of this service. The standard can be both syntax (as in WSDL) and semantic information. It is particularly important that system services be interoperable.

IP. The IP (Internet Protocol) defines an unreliable packet transfer *protocol*.

Message-oriented middleware (MOM) is software that resides in both portions of a client/server architecture and typically supports asynchronous calls between the client and server applications. Message queues provide temporary storage when the destination program is busy or not connected. http://www.middleware.org/mom/basicmom.html and http://www.sei.cmu.edu/str/descriptions/momt.html

Metadata. Metadata is information about other data to aid the understanding of such data by machines and humans alike.

Peer-to-Peer Computing. Peer-to-Peer (P2P) computing is the sharing of computer resources and services by direct exchange between systems [http://www.peer-to-peerwg.org/whatis/]. P2P systems can be used to support *virtual organizations*.

Portal. A portal is a web-accessible gateway to services.

Protocol. A protocol is a set of rules that end points in a telecommunication system use when exchanging information.

Provenance. Data provenance is a record of the process whereby a piece of data was created. This might be specified by metadata such as when the data was created, who created it, and so on. The idea of provenance can be applied more generally to workflows in which the provenance of the workflow itself (based on the provenance of the individual nodes in the workflow) and the provenance of the data output by the workflow are distinct, but related, concepts.

SDK. The term SDK (Software Development Kit) denotes a set of code designed to be linked with, and invoked from within, an application program to provide specified functionality. An SDK typically implements an *API*. If an API has multiple implementations, then there will be multiple SDKs for that API.

Semantic Grid. An infrastructure, drawing on research and development in both the Grid and Semantic Web areas, to support the full richness of the e-Science vision through a service-oriented approach [http://www.semanticgrid.org/].

Semantic Web. An extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [http://www.semanticweb.org and http://www.w3.org/2001/sw/].

Semantic interoperability. Services are semantically interoperable if the meaning of their interfaces agrees with some standard. The appropriate way to specify meaning is an active computer science research area.

Stateful and non-stateful services. None of the state of a non-stateful service instance persists beyond the lifetime of its instantiation. Some, or all, of the state of a stateful service instance persists beyond the lifetime of its instantiation.

Syntactic interoperability. Services are syntactically interoperable if the labels of their interfaces agree with some standard.

TCP. The TCP (Transmission Control Protocol) builds on *IP* to define a reliable data delivery *protocol*.

TLS. The TLS (Transport Layer Security) *protocol* defines a protocol to provide privacy and data integrity between two communicating applications. It is layered on top of a reliable transport protocol such as *TCP*.

Virtual data. Virtual data is data that can be generated on-the-fly through the execution of a *workflow*. The URI of the workflow is used as a representation of the virtual data.

Virtual Organization. A virtual organization is one that flexibly and securely shares computing resources in a controlled manner across domain boundaries in order to achieve some common aim of its members.

Virtual Private Network (VPN). A VPN is a private data network that makes use of the public telecommunications infrastructure, maintaining privacy through the use of a tunnelling protocol and security procedures [http://www.vpnc.org/vpn-technologies.pdf].

Web service. A Web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described, and discovered by XML artifacts and that supports direct interactions with other software applications using XML-based messages via internet-based protocols.

Workflow. Workflow is the automation of a process, in whole or in part, during which information or tasks are passed from one participant to another for action, according to a set of procedural rules. In the context of e-Science a "participant" is usually a *Web service* or *Grid service*.

15.2 National and Regional Programmes

ApGrid is a partnership for Grid computing in the Asia Pacific region.
http://www.apgrid.org/

DutchGrid is the platform for Grid Computing and Technology in the Netherlands. Open to all institutions for research and test-bed activities, the goal of DutchGrid is to coordinate the various deployment efforts and to offer a forum for the exchange of experiences on Grid technologies.
http://www.dutchgrid.nl

Esnet (Energy Sciences Network) is a high-speed network serving thousands of U.S. Department of Energy scientists and collaborators worldwide. ESnet enables researchers at national laboratories, universities and other institutions to communicate with each other using the collaborative capabilities needed to address some of the world's most important scientific challenges.
http://www.es.net/

INFN-Grid is the focal point for Italian Grid activities. Established several years ago, the Italian National Grid Initiative is one of the major initiatives in Europe involving researchers from across Italy.
http://www.infn.it/grid

The NMI (NSF Middleware Initiative) aims to help scientists and researchers use the Internet to effectively share instruments, laboratories and data, and to collaborate with their colleagues. NMI will create and deploy advanced network services for simplifying access to diverse Internet resources.
http://www.nsf-middleware.org/

NORDUGRID is establishing an inter-Nordic test bed facility for the implementation of wide area computing and data handling. In order to set up such a test bed, existing Grid middleware tools are being extended and modified to suit distributed applications. The facility is providing the infrastructure for interdisciplinary feasibility studies of GRID-like computer structures across the Nordic countries.
http://www.nordugrid.org

The UK e-Science Programme is Europe's largest Grid initiative. It is structured around six key elements – a National e-Science Centre linked to a network of Regional Centres; a Grid Core Programme – developing key Grid middleware components – and Demonstrator Projects; Grid IRC Research Projects; Support for e-Science Testbeds; International standards setting; and Grid Networking. The programme is distributed across the seven UK Research Councils and is additionally supported by the UK Department of Trade & Industry.
http://www.escience-grid.org.uk and http://www.rcuk.ac.uk/escience/

The W3C (World Wide Web Consortium) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding.
http://www.w3.org/

15.3 Descriptions of Projects

Akenti is a security model and architecture that is intended to provide scalable security services in highly distributed network environments. The project aims to achieve the same level of expressiveness of access control that would be accomplished through a local human controller in the decision loop, and to accurately reflect the existing policy, authority, delegation, and responsibility present in these environments.
http://www-itg.lbl.gov/Akenti/

The AKT (Advanced Knowledge Technologies) programme is based around six challenges to ease fundamental bottlenecks in the engineering and management of knowledge. These are acquiring knowledge, modelling knowledge, retrieving knowledge, reusing knowledge, publishing knowledge and maintaining knowledge.
http://www.aktors.org/akt/

The ALICE (A Large Ion Collider at CERN) Collaboration is building a dedicated heavy-ion detector to exploit the unique physics potential of nucleus-nucleus interactions at LHC energies. The aim is to study the physics of strongly interacting matter at extreme energy densities, where the formation of a new phase of matter, the quark-gluon plasma, is expected.
http://alice.web.cern.ch/Alice/

The goal of AstroGrid is to build a grid infrastructure that will allow the construction of a Virtual Observatory, unifying the interfaces to astronomy databases and providing remote access as well as assimilation of data. The Virtual Observatory is a truly global problem and AstroGrid will be the UK contribution to the global Virtual Observatory.
http://www.astrogrid.org/

The BIOGRID project is conducting a trial for the introduction of the Grid approach in the biotechnology industry. The project is focussing on the integration of three existing technologies – agent technology, automatic model classification technology and knowledge visualisation technology – and the production of a working prototype biotechnology information Grid.
http://www.bio-grid.net

Cactus is an open source problem solving environment designed for scientists and engineers. Its modular structure easily enables parallel computation across different architectures and collaborative code development between different groups.
http://www.cactuscode.org

The CCLRC Data Portal project aims to develop the means for a scientist to explore these data resources, discover the data they need and retrieve the relevant datasets through one interface and independent of the data location. The CLRC DataPortal enables the parallel search and exploration of distributed heterogeneous metadata catalogues as well as providing access to the data itself.
http://ws1.esc.rl.ac.uk/web/projects/dataportal

The CCLRC Data Portal trial access scheme currently allows access to selected metadata and data from four facilities: the Synchrotron Radiation Department (SRD), the Neutron Spallation Source (ISIS), the British Atmospheric Data Centre (BADC), and the Max-Planck Institute for Meteorology (MPIM).
http://esc.dl.ac.uk:9000/dataportal/index.html

The CCLRC HPC Portal project aims to develop a generic Grid Services Portal for the UK HPC community.
http://esc.dl.ac.uk/HPCPortal/

The CEOS (Committee on Earth Observation Satellites) is an international organization charged with coordinating international civil spaceborne missions designed to observe and study planet Earth.
http://www.ceos.org/

The CHEF (CompreHensive collaborativE Framework) initiative has as its goal, the development of a flexible environment for supporting distance learning and collaborative work, and doing research on distance learning and collaborative work.
http://www.chefproject.org/index.htm

The CoABS (Control of Agent-Based System) programme develops and evaluates a wide variety of alternative agent control and coordination strategies to determine the most effective strategies for achieving the benefits of agent-based systems, while assuring that self-organizing agent systems will maintain acceptable performance and security protections.
http://www.darpa.mil/ipto/research/coabs/ and http://coabs.globalinfotek.com/

The CoABS Grid is middleware that integrates heterogeneous agent-based systems, object-based applications, and legacy systems. It includes a method-based application programming interface to register agents, advertise their capabilities, discover agents based on their capabilities, and send messages between agents. CoABS Grid is part of the Control of Agent-Based Systems programme.
http://coabs.globalinfotek.com/public/Grid/What_is_Grid.htm and http://coabs.globalinfotek.com/

The CoAKTinG (Collaborative Advanced Knowledge Technologies in the Grid) project aims to advance the state of the art in collaborative mediated spaces for distributed e-Science collaboration through the novel application of advanced knowledge technologies.
http://www.aktors.org/coakting/

The Comb-e-Chem project is developing an integrated platform that combines existing structure and property data sources within a Grid-based information-and knowledge-sharing environment concerned with the synthesis of new compounds by combinatorial methods.
http://www.combechem.org/

The Computational Markets project will design and implement facilities for pricing, accounting and charging for all types of Grid service (software, hardware, data, network capacity, etc.). These trading mechanisms will be implemented as extensions to the *OGSA* and its reference implementation the Globus Toolkit 3.
http://www.lesc.ic.ac.uk/markets/

The Condor project supports high throughput computing on large collections of distributively owned resources. Condor is a specialized workload management system for compute-intensive jobs. Condor provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their serial or parallel jobs to Condor, Condor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.
http://www.cs.wisc.edu/condor/

CORBA (Common Object Request Broker Architecture) is a standard infrastructure for distributed computing, and is widely-used in industrial and commercial applications.
http://www.corba.org/

The Corporate Ontology Grid (COG) project is demonstrating the commercial application of Grid technologies via the use of ontological modelling to integrate corporate information. It hopes to realise the concept of an information Grid incorporating real corporate data. It is focussing on giving semantics to corporate data formats and the automatic translation between formats via a semantic mapping to a central ontology which will be published and reusable.
http://www.cogproject.org

CrossGrid is developing technologies for large-scale Grid-enabled simulations and visualisations that require responses in real-time. The project addresses issues such as the distribution of source data, simulation and visualisation, virtual time management, interactive simulation and visualisation rollback and platform-independent virtual reality.
http://www.eu-crossgrid.org

The DAME (Distributed Aircraft Maintenance Environment) project is building a Grid-based distributed diagnostics system for aircraft engines. The project addresses performance issues such as large-scale data management with real time demands.
http://www.cs.york.ac.uk/dame/

DAMIEN is developing essential software so that the Grid can be used for industrial simulation and visualisation. DAMIEN builds on existing tools and libraries. DAMIEN will develop a set of utilities which will enable developers to port their applications more easily to the Grid.
http://www.hlrs.de/organization/pds/projects/damien

DataTAG is implementing a network infrastructure for a truly high-speed interconnection between individual Grid domains both in Europe and the US. The project incorporates the design and implementation of advanced network services for guaranteed data delivery, transport protocol optimisation, efficiency and reliability of network resource utilisation, user-perceived application performance and middleware interoperability across different domains.
http://datatag.web.cern.ch/datatag

The seminar on "Digital Curation: Digital Archives, Libraries, and e-Science", sponsored by the Digital Preservation Coalition and the British National Space Centre, aimed to raise the profile of the Open Archival Information System Reference Model (OAIS) standard in the UK and share practical experience of digital curation in the digital library sector, archives, and e-sciences.
http://www.dpconline.org/graphics/events/digitalarchives.html

The Discovery Net project is designing, developing and implementing an advanced infrastructure to support real-time processing, interpretation, integration, visualization and mining of massive amounts of time critical data generated by high throughput devices. The project covers new technology devices and technology including biochips in biology, high throughput screening technology in biochemistry and combinatorial chemistry, high throughput sensors in energy and environmental science, remote sensing and geology.
http://www.discovery-on-the.net/

DOE ScienceGrid links the distributed compute, data, and instrument resources of the U.S. Department of Energy research laboratories.
http://doesciencegrid.org/ and
http://doecollaboratory.pnl.gov/research2/doesciencegrid/2pager_march2003.pdf

EUROGRID is developing core Grid software components and integrating them into an environment providing fast file transfer, resource brokerage, interfaces for coupled applications and interactive access. This environment will be developed and tested against the requirements of applications in biochemistry, weather forecasting, computer-aided engineering, structural analysis and real-time data processing.
http://www.eurogrid.org

The EAVO (European Astrophysical Virtual Observatory) is designing and implementing a European virtual observatory. A virtual observatory is a collection of interoperating data archives and software tools. These are connected by the Internet to form an environment for astronomical research.
http://www.eso.org/projects/avo

eDIKT is a project funded through a Research Development Grant by the Scottish Higher Education Funding Council, which applies solid software engineering techniques to leading-edge computer science research to produce robust, scalable data management tools that enable new research areas in e-Science. It has major activities extending OGSA-DAI.
http://www.edikt.org/

The e-Materials project is using a combination of novel computational and computer science methodologies and teams will be used to develop GRID e-Science technologies to deliver new simulation solutions to problems and fields relating to combinatorial materials science and polymorph prediction.
http://www.e-science.clrc.ac.uk/web/projects/complexmaterials

The e-Minerals project brings together simulation scientists, applications developers and computer scientists to develop UK e-science/Grid capabilities for molecular simulations of environmental issues. A common set of simulation tools is being developed for a wide range of applications, and the Grid environment will be established which will result in a giant leap in the capabilities of these powerful scientific tools.
http://eminerals.org/

The European DataGrid (EDG) is enabling next generation scientific exploration that requires intensive computation and the analysis of shared, large-scale databases. These databases range in size from hundreds of TeraBytes to PetaBytes and are used by widely distributed scientific communities. The project is devising and developing scalable software solutions and testbeds to handle many PetaBytes of distributed data, tens of thousand of computing resources including processors, disks, other devices and thousands of simultaneous users from collaborating research institutes.
http://www.edg.org

The European Grid of Solar Observations (EGSO) is laying the foundations of a virtual solar observatory. EGSO addresses the problem of combining heterogeneous data from scattered archives of space- and ground-based observations into a single “virtual” dataset. A new, unified solar feature catalogue, based on a simple user interface, will allow the user to search for observations on the basis of events and phenomena, rather than just time and location.
http://www.egso.org

The ESG (Earth System Grid) II project is addressing the formidable challenges associated with enabling analysis of, and knowledge development from, global Earth System models. Through a combination of Grid technologies and emerging community technology, distributed federations of supercomputers and large-scale data and analysis servers will provide a seamless and powerful environment that enables the next generation of climate research.
http://www.earthsystemgrid.org/

FLOWGRID is an example of a Grid computing environment project, with the vision of revolutionizing the way Computational Fluid Dynamics (CFD) simulations are set up, executed and monitored on geographically and organisationally dispersed computing resources. Computational Grids suit CFD simulations because resources for CFD simulations are transient. With the Grid, organisations that use CFD will be able to access computation and information resources on demand.
http://www.unizar.es/flowgrid

The Gateway project is investigating the use of distributed computing, computational grid, and Internet protocols and technologies to support scientific computing and to build secure computational web portals. Particular research areas include distributed computing with Web services, distributed security, and client environments for the Grid.
http://www.gatewayportal.org/

The GEMMS (Grid-Enabled Medical Simulation Services) project is demonstrating how the Grid can be used to transform healthcare and enable Europe to lead that transformation. The GEMSS testbed will render accessible a multitude of medical computing and resource services in a clinical environment – including new tools for improved diagnosis, operation planning, and surgical procedures.
http://www.ccrl-nece.de/gemss

The GEODISE (Grid-Enabled Optimisation and Design Search for Engineering) project provides Grid-based seamless access to an intelligent knowledge repository, a state-of-the-art collection of optimisation and search tools, industrial strength analysis codes, and distributed computing and data resources. The aim is to exploit engineering modeling and analysis to yield improved engineering designs.
http://www.geodise.org/

The GGF (Global Grid Forum) is a community-initiated forum of thousands of individual researchers and practitioners working on distributed computing, or "grid" technologies. GGF's primary objective is to promote and support the development, deployment, and implementation of Grid technologies and applications via the creation and documentation of "best practices" – technical specifications, user experiences, and implementation guidelines.
http://www.gridforum.org/

The Globus project is developing fundamental technologies needed to build computational grids.
http://www.globus.org

The GODIVA (Grid for Ocean Diagnostics, Interactive Visualisation and Analysis) project is developing a prototype ocean diagnostics Grid for community work with high resolution marine data from the NERC OCCAM model and data from the Met Office FOAM (Forecasting Ocean-Atmosphere Model). The proposal will also develop new software for more sophisticated remote visualisation capabilities of large amounts of environmental data created in oceanic and atmospheric science, including the above models.
http://www.e-science.clrc.ac.uk/web/projects/godiva

GRACE is developing a distributed search and categorisation engine on top of Grid technology. It aims to make terabytes of unstructured textual information distributed across a vast number of geographically distant locations highly accessible. Such information is typically handled by search engines which are almost all extremely centralised. In order to index a document they must download it, process it and store its index - all in one central location. GRACE addresses situations in which a centralized index is simply unfeasible.
http://www.grace-ist.org

The GrADS (Grid Application Development Software) project is to simplify distributed heterogeneous computing in the same way that the World Wide Web simplified information sharing over the Internet. The GrADS project is exploring the scientific and technical problems that must be solved to make grid application development and performance tuning for real applications an everyday practice.
http://www.hipersoft.rice.edu/grads/

The GRASP (GRid Application Service Provision) project is studying, designing, developing and validating a new advanced system infrastructure for Application Service Provision based on Grid technologies combined with commodity technologies (such as Microsoft .NET, Web Services, XML, SOAP, etc.).
http://www.eu-grasp.net

The GRIA (Grid Resources for Industrial Applications) project aims to make the Grid usable by business and industry. It is devising business models and processes that make it feasible and cost-effective to offer and use computational services securely in an open Grid marketplace. The project is implementing enabling technologies for this including business process (workflow) enactment tools, quality of service prediction and management, and negotiation semantics and agent implementations through which users can discover, buy access to and bind with application services.
http://www.gria.org

The Gridbus project is concerned with the design and development of next-generation computing systems and applications that aggregate or lease services of distributed resources depending on their availability, capability, performance, cost, and users' quality-of-service requirements.
http://www.cs.mu.oz.au/~raj/grids/

GridLab is developing software to enable simulation and visualisation codes to adapt to changing Grid environments and to be able fully to exploit dynamic resources. It supports the construction of applications that can migrate from site to site during their execution both in whole or in part, to spawn related tasks and acquire or release resources depending on the changing availability of Grid resources and on the needs of the applications themselves.
http://www.gridlab.org

The GridOneD project is concerned with the investigation of technologies to create middleware components to help software developers and application users to utilize the Grid infrastructure. The main areas of interest are to support developers and users integrate legacy applications; to provide a data-mediation layer for data-type translation between applications; to provide resource discovery and resource monitoring tools; to optimize resource selection and usage, based on user-specified criteria.
http://www.gridoned.org/

The goal of the Grid Portal Architecture Workshop was to articulate a community standard architecture for Grid Portals. This portal architecture would allow portal application developers to build specialized application components (portlets) that could be easily installed in any compliant portal server and easily configured by users to operate in their environment. This architecture should provide a standard way for the application portlets to share information and important user specific context data such as proxy certificates.
http://www.computingportals.org/meetings/ggf7/

GRIP (GRid Interoperability Project) is a 2 year research project, funded in part by the European Union (EU) to realise the interoperability of Globus and UNICORE and to work towards standards for interoperability in the Global Grid Forum.
http://www.grid-interoperabilty.org

The GriPhyN (Grid Physics Network) project is developing tools and software infrastructure for petabyte-scale data intensive science. The project is based around the data requirements of four key experiments: the Compact Muon Soleniod (CMS) and ATLAS experiments at the LHC; LIGO (Laser Interferometer Gravitational-wave Observatory); and SDSS (Sloan Digital Sky Survey).
http://www.griphyn.org/.

GridPort (the Grid Portal Toolkit) is a collection of technologies designed to aid in the development of science portals on computational grids.
https://gridport.npaci.edu/index.html

The GridPP project is delivering Grid middleware and hardware infrastructure to enable testing of a prototype of the Grid for the Large Hadron Collider (LHC) project at CERN. GridPP is closely related with the European Data Grid project.
http://www.gridpp.ac.uk/

GridSolve: please see NetSolve.

GRIDSTART is a project clustering initiative with the specific objective of consolidating Grid technical advances in Europe, encouraging interaction amongst similar activities both in Europe and the rest of the world and stimulating the early take-up by industry and research of Grid-enabled applications.
http://www.gridstart.org

Grid Systems is a company that markets software for aggregating the computing resources in an organization to supply high performance to applications. Its main product is InnerGrid.
http://www.gridsystems.com/

The GridWeaver project will investigate and propose solutions to the problem of automating the configuration and management of Grid computing fabrics. The goal of this project is to bring together relevant insights, experience and existing technologies from two different domains: configuration management for large-scale computing fabrics, and configuration, deployment and management of large-scale distributed applications.
http://www.epcc.ed.ac.uk/gridweaver/

Groove Networks is a company that markets desktop collaboration software that can be used to create secure project-specific virtual shared spaces for working with different groups of people, and that also supports file sharing.
http://www.groove.net/

The gViz project is extending IRIS Explorer and its collaborative tools to work in a Grid environment, and will develop a fully Grid-enabled extension of IRIS Explorer, to allow an e-scientist to run Grid applications through the IRIS Explorer user interface.
http://www.visualization.leeds.ac.uk/gViz/

The HPC-VAO project developed an environment for vibro-acoustic optimisation. This is a computationally complex process involving structural analysis to find the natural vibration frequencies of the mechanical system being optimised, and then an acoustic analysis to find the noise levels at the locations of interest.
http://www.it-innovation.soton.ac.uk/research/grid/hpcvao.shtml

The goal of the ICENI project is to provide high-level abstractions for e-Science which will allow users to construct and define their own applications through a graphical composition tool integrated with distributed component repositories and to deliver this environment across a range of platforms and devices.
http://www.lesc.ic.ac.uk/iceni/

The Internet2 E2Epi (end-to-end performance initiave) aims to create a predictable, and well-supported environment in which Internet2 campus network users have routinely successful experiences in their development and use of advanced Internet applications by focusing resources and efforts on improving performance problem detection and resolution throughout campus, regional, and national networking infrastructures.
http://e2epi.internet2.edu/

The iVDGL (International Virtual Data Grid Laboratory) is a global Data Grid that will serve forefront experiments in physics and astronomy. Its computing, storage and networking resources in the U.S., Europe, Asia and South America provide a unique laboratory that will test and validate Grid technologies at international and global scales. Sites in Europe and the U.S. will be linked by a multi-gigabit per second transatlantic link funded by the European *DataTAG* project.

http://www.ivdgl.org

The IVOA (International Virtual Observatory Alliance) is an alliance of various virtual observatory projects, such as *AstroGrid*, *EAVO*,and *NVO*, that is pursuing an international virtual observatory and the expansion of astronomical research capabilities.
http://www.ivoa.net/

The LCG (LHC Computing Grid) project is providing the resources needed to satisfy the computational requirements of experiments with the Large Hadron Collider (LHC). A key feature is the large rate of data production (tens of petabytes per year). These computing needs will be met by deploying a worldwide computational grid service, integrating the capacity of scientific computing centres spread across Europe, America and Asia into a virtual computing organisation.
http://lcg.web.cern.ch/LCG/

Legion is an object-based metasystems software project designed for a system of millions of hosts and trillions of objects tied together with high-speed links. Users working on their home machines see the illusion of a single computer, with access to all kinds of data and physical resources, such as digital libraries, physical simulations, cameras, linear accelerators, and video streams.
http://legion.virginia.edu/

The Liberty Alliance aims to establish an open standard for federated network identity through open technical specifications.
http://www.projectliberty.org/

MAMMOGRID aims to develop a Europe-wide database of mammograms that will be used to investigate a set of important healthcare applications as well as the potential of the Grid to support effective co-working between healthcare professionals throughout the EU.

The MIAKT (Medical Imaging and Advanced Knowledge Technologies) project is developing collaborative medical problem solving using knowledge services provided via the e-Science Grid infrastructure. The work focuses on image and signal interpretation and the use of complex data in decision support. The initial focus of the project will be on Triple Assessment in symptomatic focal breast disease.
http://www.aktors.org/miakt/

The Mono project is an effort to create an open source implementation of the .NET Development Framework. Mono includes: a compiler for the C# language, a runtime for the Common Language Infrastructure (also referred as the CLR) and a set of class libraries. The runtime can be embedded into your application. Mono has implementations of both ADO.NET and ASP.NET as part of its distribution.
http://www.go-mono.com/

MOSES is developing a modular and scalable ontology-based knowledge management system, through the customisation and integration of several technologies – software agents, NLP tools and graph theory. The effectiveness of this system and the viability of the approach will be demonstrated by an application supporting ontology-based queries expressed in natural language in the domain of academic Web sites and a set of semi-automatic tools showing how new content may be easily added to the knowledge structure.
http://www.hum.ku.dk/moses

The MyGrid project is to design, develop and demonstrate higher level functionalities over an existing Grid infrastructure that support scientists in making use of complex distributed resources. An e-Scientist's workbench is being developed to support: the scientific process of experimental investigation, evidence accumulation and result assimilation; the scientist's use of the community's information; and scientific collaboration, allowing dynamic groupings to tackle emergent research problems.
http://mygrid.man.ac.uk/

The NASA Information Power Grid (IPG) is a high-performance computing and data grid. Grid users can access widely distributed heterogeneous resources from any location, with IPG middleware adding security, uniformity, and control.
http://www.ipg.nasa.gov/

The NEESGrid (National Earthquake Engineering and Simulation Grid) is a distributed virtual laboratory for advanced earthquake experimentation and simulation. The aim is to develop a national resource for research and education supporting simulation, collaborative experimentation, and modeling for the earthquake engineering community. NEESgrid will shift the emphasis of earthquake engineering research from reliance on physical testing to integrated experimentation, computation, theory, and databases.
http://www.neesgrid.org/

The NDG (NERC DataGrid) will make data discovery, delivery and use much easier than it is now, facilitating better use of the existing investment in the curation and maintenance of quality data archives. Although it will initially concentrate on oceanographic and atmospheric data, the technology plan is aiming at supporting data from the wider environmental community.
http://ndg.nerc.ac.uk/

The NetSolve project (now named GridSolve) aims to bring together distributed, heterogeneous, computational resources connected by computer networks for solving complex scientific problems. NetSolve is an RPC based client/agent/server system that allows one to remotely access both hardware and software components.
http://www.cs.utk.edu/netsolve/

The Network Weather Service (NWS) is a distributed system that periodically monitors and dynamically forecasts the performance that various network and computational resources can deliver over a given time interval. The service operates a distributed set of performance sensors (network monitors, CPU monitors, etc.) from which it gathers readings of the instantaneous conditions. It then uses numerical models to generate forecasts of what the conditions will be for a given time frame.
http://nws.cs.ucsb.edu/

The Ninf project is developing Grid technologies that allow users to access various resources such as hardware, software and scientific data on the Grid with an easy-to-use interface. The Ninf system is a Grid RPC and provides RPC facilities designed to provide a programmming interface similar to conventional function calls and enable the user to build Grid-enabled applications.
http://ninf.apgrid.org/

Ninf-G is a reference implementation of a Grid RPC (Remote Procedure Call) system using the Globus Toolkit.
http://ninf.apgrid.org/documents/ng/

The NVO (National Virtual Observatory) collaboration aims to create standards for astronomical data collections that will be used by the wide astronomical community. In this way data will be easier to use, easier to find, and easier to join with other data. A second thrust is exploring the use of high-performance computing resources for discovery in astronomy.
http://www.us-vo.org/

The OGSA-DAI (Open Grid Services Architecture Database Access and Integration) project is concerned with constructing middleware to assist with access and integration of data from separate data sources via the grid. It is engaged in identifying the requirements, designing solutions and delivering software that will meet this purpose.
http://www.ogsadai.org.uk/ and http://www.gridforum.org/6_DATA/dais.htm

The OGSI.net project at the University of Virginia is dedicated to creating clients and services that are compatible with the Open Grid Services Infrastructure specification.
http://www.cs.virginia.edu/~gsw2c/ogsi.net.html

The Open Archives Initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content. The Open Archives Initiative has its roots in an effort to enhance access to e-print archives as a means of increasing the availability of scholarly communication.
http://www.openarchives.org/

OPENMOLGRID addresses large scale molecular design problems in a Grid environment. The use of the Grid is essential as the amount of data is huge and model building is computationally very demanding. The project is making use of the EUROGRID infrastructure for adaptation of existing software modules, and to make a solid foundation for the resulting molecular engineering tools. These tools will then be used to develop a prototype application for the generation molecular structures with given chemical properties or biological activities.
http://www.openmolgrid.org

Parabon Computation is a company that develops software for harnessing the unused capacity of Internet-connected computers and delivering it to the desktops of those who need supercomputing.
http://www.parabon.com/

The PERMIS (Privilege and Role Management Infrastructure Standards Validation) project explores and demonstrates the feasibility of the distributed approach to authorisation and authentication in which each attribute owner or manager directly certifies the attributes of individuals. The PERMIS project has designed and built a completely new infrastructure, a so called "Privilege Management Infrastructure" (PMI) that can be defined as the complete set of processes required to provide an authorisation service.
http://www.permis.org/

PlaceWare is a Microsoft company that markets Web conferencing software.
http://www.placeware.com/

The PPDG (Particle Physics Data Grid) collaboration focuses on data grid services to enable the worldwide distributed computing model of current and future high-energy and nuclear physics experiments.
http://www.ppdg.net/

The PRISM (Program for Integrated Earth System Modelling) is an infrastructure project funded by the European Commission. Its main objective is to provide a software infrastructure facilitating the assembly, execution and post-processing of Earth System model simulations, based on existing state-of-the-art European Earth System components models.
http://prism.enes.org/main.html

The PROMENVIR project developed a stochastic modelling environment, allowing engineers to incorporate the effect of uncertainties into models of mechanical systems, which has important benefits in analysis-led design and product development processes.
http://www.it-innovation.soton.ac.uk/research/grid/promenvir.shtml

Pythia-II proposes an approach for dealing with the task of locating solution software for problems, and then selecting from among many alternatives by processing performance data from the targeted software. The high level of complexity involved in the algorithmic discovery of knowledge from performance data and in the management of the performance data and discovered knowledge is handled by combining knowledge discovery in databases (KDD) methodologies, and recommender system technologies.
http://www.cs.purdue.edu/research/cse/pythia/index.html

The RealityGrid project integrates high performance computing and visualisation facilities providing a synthetic environment for modelling complex condensed matter systems at the molecular and mesocale levels. These models will be compared and integrated with experimental data.
http://www.realitygrid.org

The RGR (Relational Grid Resource) project explores the representation of grid resource data, including information and attributes about resources in a computational or data grid, including compute servers, clusters, storage resources, people, groups of people, and so on.
http://www.cs.indiana.edu/~plale/projects/RGR/

The SCEC Grid project is developing computing capabilities that will lead to better forecasts of when and where earthquakes are likely to occur in southern California, and how the ground will shake as a result. The project will develop the ability for scientists to improve computer models of how the Earth is structured and how the ground moves during earthquakes.
http://iowa.usc.edu/cmeportal/index.html

The Self e-Learning Networks project (SeLeNe) is conducting a study into the technical feasibility of using Semantic Web technology for dynamically integrating metadata from heterogeneous and autonomous educational resources, and for creating personalised views over this Knowledge Grid.
http://www.dcs.bbk.ac.uk/~ap/projects/selene

The ServoGrid (Solid Earth Research Virtual Observatory Grid) addresses the critical need for seamless access to large distributed volumes of data in Earth Sciences.
http://www.servogrid.org/

SETI@home is a scientific experiment that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI).
http://setiathome.ssl.berkeley.edu/

The Shibboleth project is developing architectures, policy structures, practical technologies, and an open source implementation to support inter-institutional sharing of web resources subject to access controls. In addition, Shibboleth is developing a policy framework that will allow inter-operation within the higher education community.
http://shibboleth.internet2.edu/

The Jakarta Slide project is composed of multiple modules tied together using *WebDAV*. The main module is a Content Management and Integration System, which can be seen as a low-level content management framework. Conceptually, it provides a hierarchical organization of binary content which can be stored into arbitrary, heterogenous, distributed data stores. In addition, Slide integrates security, locking, versioning, as well as many other services.
http://jakarta.apache.org/slide/

TeraGrid is a multi-year effort to build and deploy the world's largest, fastest, distributed infrastructure for open scientific research. When completed, the TeraGrid will include 20 teraflops of computing power distributed at five sites, facilities capable of managing and storing nearly 1 petabyte of data, high-resolution visualization environments, and toolkits for grid computing. These components will be tightly integrated and connected through a network that will operate at 40 gigabits per second—the fastest research network on the planet.
http://www.teragrid.org/

TERENA (Trans-European Research and Education Networking Association) was to promote and participate in the development of a high quality international information and telecommunications infrastructure for the benefit of research and education in Europe.
http://www.terena.nl/ and http://www.terena.nl/tech/tac/zagreb/TAC-200305.pdf

Triana is a graphical programming environment that allows users to create complex programs out of basic building blocks supplied with the system by dragging the blocks, dropping them into a workspace, and connecting them together.
http://www.triana.co.uk/ and http://www.trianacode.org/

Trillium aims to interlink three major Grid projects: the Particle Physics Data Grid (*PPDG*), the International Virtual Data Grid (*IVDG*) and the Grid Physics Network (*GriPhyN*).
http://www.hicb.org/TrilliumNewsletter/index.htm

UNICORE (UNiform Interface to COmputing REsources) allows users to submit jobs to remote high performance computing resources without having to learn details of the target operating system, data storage conventions and techniques, or administrative policies and procedures at the target site.
http://www.fz-juelich.de/unicore/

The UNICORE Forum promotes the use of high performance computers in science and research, and ensures that authorised scientists at German research institutions and universities have easy and secure access from their workstation to national high performance computers through the use of *UNICORE*.
http://www.unicore.org/forum.htm

The UNICORE Plus project has developed a grid infrastructure together with a computing portal for engineers and scientists to access supercomputer centers from anywhere on the Internet. This has to be done with strong authentication in a uniform and easy to use way. The differences between platforms will be hidden from the user thus creating a seamless interface for accessing supercomputers, compiling and running applications, and transferring input/output data.
http://www.fz-juelich.de/unicoreplus/index.html

UNICOREpro provides a seamless interface for preparing jobs, submitting them to remote computing resources, and monitoring their process. Workflows can include data transfers, execution of applications, and compile/link steps. Specialized interfaces can be integrated for important applications.
http://www.pallas.com/e/products/unicore-pro/index.htm

15.4 Descriptions of Systems and Tools

The Access Grid is an ensemble of resources including multimedia large-format displays, presentation and interactive environments, and interfaces to Grid middleware and to visualization environments. These resources are used to support group-to-group interactions across the Grid.
http://www.accessgrid.org/

AliEn (Alice Environment) is a Grid prototype created by the Alice Offline Group. AliEn has includes the following: Distributed Catalogue, Authentication Server, Queue Server, Computing Elements, Storage Elements, Information Server.
http://alien.cern.ch/

Apache Axis is an implementation of SOAP 3.0 and can be used to support the installation and deployment of Web services.
http://ws.apache.org/axis/

The Apache Web Services Invocation Framework (WSIF) is a simple Java API for invoking Web services. WSIF enables developers to interact with abstract representations of Web services through their WSDL descriptions instead of working directly with the Simple Object Access Protocol (SOAP) APIs. WSIF allows stubless or completely dynamic invocation of a Web service, based upon examination of the meta-data about the service at runtime.
http://ws.apache.org/wsif/

AVS/Express is a visualization tool that provides powerful visualization methods for challenging problems in fields such as science, business, engineering, medicine, telecommunications and environmental research. AVS/Express combines rapid data analysis and rich visualization techniques with a graphical application development environment.
http://www.avs.com/software/soft_t/avsxps.html

The Biology WorkBench is a web-based tool for biologists. The WorkBench allows biologists to search many popular protein and nucleic acid sequence databases. Database searching is integrated with access to a wide variety of analysis and modeling tools, all within a point and click interface that eliminates file format compatibility problems.
http://workbench.sdsc.edu/

BPEL4WS (Business Process Execution Language for Web Services) provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions.
http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

BPWS4J (Business Process Execution Language for Web Services Java Run Time) from IBM includes a platform upon which can be executed business processes written using the *BPEL4WS*, a set of samples demonstrating the use of *BPEL4WS*, and a tool that validates *BPEL4WS* documents.
http://www.alphaworks.ibm.com/tech/bpws4j

Capital Radio is the UK's leading commercial radio group, and is considering peer-to-peer solutions to its data mobility and replication problems.
http://www.capitalradiogroup.com/

CAS (Community Authorization Service) allows resource providers to specify coarse-grained access control policies in terms of communities as a whole, delegating fine-grained access control policy management to the community itself. Resource providers retain ultimate authority over their resources but are spared day-to-day policy administration tasks.
http://www.globus.org/security/CAS/

The CCA (Common Component Architecture) defines a minimal set of standard interfaces that a high-performance component framework has to provide to components, and can expect from them, in order to allow disparate components to be composed together to build a running application.
http://www.cca-forum.org/

CERN is the European Organisation for Nuclear Research, the world's largest particle physics centre. Here physicists come to explore what matter is made of and what forces hold it together. CERN exists primarily to provide them with the necessary tools. These are accelerators, which accelerate particles to almost the speed of light and detectors to make the particles visible.
http://public.web.cern.ch/public/

The Chimera Virtual Data System (VDS) provides a mechanism for producing a given logical data file in the form of an abstract program execution graph.
http://www.griphyn.org/chimera/

ClassAds (Classified Advertisements) are used by Condor for describing jobs, workstations, and other resources. In addition, they are exchanged by Condor processes to schedule jobs, they are logged to files for statistical and debugging purposes, and they are used to enquire about current state of the system.
http://www.cs.wisc.edu/condor/classad/

CoAX (Coalition Agents experiment) is an international collaboration carried out under the auspices of DARPA's Control of Agent-Based Systems (*CoABS*) program. Building on the CoABS Grid framework, the CoAX agent infrastructure groups agents into domains that reflect real-world organizational, functional, and national boundaries, such that security and access to agents and information can be governed by policies at multiple levels.
http://www.aiai.ed.ac.uk/project/coax

Condor-G allows users to harness multidomain resources as if they belong to one personal domain. It is based on *Globus* and *Condor* software.

http://www.cs.wisc.edu/condor/condorg/

A Condor glide-in allows the temporary addition of a Globus resource to a local Condor pool. This is accomplished by installing and executing some of the Condor daemons on the Globus resource.
http://www.cs.wisc.edu/condor/manual/v6.4/condor_glidein.html

The CORBA 3.0 Notification Service is based on, and extends, the CORBA Event Service by adding new functionality and interfaces, in particular to support Quality-of-Service.
http://www-106.ibm.com/developerworks/webservices/library/co-cjct8/?dwzone=webservices

The Core Grid Functions of the Grid Protocol Architecture Working Group of the GGF define the current practice for a minimal set of Grid functions that provide uniform interfaces to architecturally, geographically, and administratively heterogeneous computing, data, and instrument systems that are managed by production Grids.
http://www.gridforum.org/Meetings/ggf7/drafts/CoreGridFunctions.v3.1.Word95.doc

Covisa-G is a demonstrator funded by the UK e-Science programme that has been developed using the Iris Explorer modular visualization system. This demonstrator enables a team of scientists to work together over the Internet to collaboratively steer and visualize a simulation. The demonstrator has been extended using Globus to run on a computational grid, whereby the simulation can be run on a Grid resource but the visualization and steering handled by scientists at their desktops.
http://www.visualization.leeds.ac.uk/CovisaG/

DAGMan (Directed Acyclic Graph Manager) is a meta-scheduler for *Condor*. It manages dependencies between jobs at a higher level than the Condor scheduler.
http://www.cs.wisc.edu/condor/dagman/

DIAMANT is a user-centric environment for restoring degraded film using a suite of compute- and data-intensive digital video processing techniques. To provide the necessary computing power, data storage and visualisation bandwidth, DIAMANT uses a Grid resourcing model to locate the necessary resources on a Linux Beowulf cluster.
http://www.it-innovation.soton.ac.uk/research/grid/diamant.shtml

The DMF (Distributed Monitoring Framework) aims to improve end-to-end throughput for data intensive applications in high speed WAN (Wide Area Network) environments, and to provide the ability to do performance analysis and fault detection in a Grid computing environment. This monitoring framework will provide accurate, detailed, and adaptive monitoring of all distributed computing components, including the network.
http://www-didc.lbl.gov/DMF/

DPML (Discovery Process Markup Language) is an XML-based language that describes processes as dataflow graphs. Nodes representing data sets are composed as a directed acyclic graph whose edges determine where each node's output should be piped.
http://jameel.recoil.org/publications/allhands2002.pdf

The DPSS (Distributed Parallel Storage System) is a data block server that provides high-performance data handling and an architecture for building high-performance storage systems from low-cost commodity hardware components.
http://www-didc.lbl.gov/DPSS/

The EBI (European Bioinformatics Institute) is a non-profit academic organisation that forms part of the European Molecular Biology Laboratory (EMBL). The EBI is a centre for research and services in bioinformatics. The Institute manages databases of biological data including nucleic acid, protein sequences and macromolecular structures.
http://www.ebi.ac.uk/

EcoGrid research focuses on the development of economic or market-based resource management and scheduling system for global grid computing.
http://www.cs.mu.oz.au/~raj/grids/ecogrid/

The EDG HEPCAL (High Energy Physics Common Application Layer) was intended to define a common application layer for LHC experiments by examining common use case.
https://edms.cern.ch/file/375586/1.3/HEPCAL_finalreport.doc

The EDG Information and Monitoring Service work package (WP3) aims to specify, develop, integrate and test tools and infrastructure to enable end-user and administrator access to status and error information in a Grid environment and to provide an environment in which application monitoring can be carried out. This will permit both job performance optimisation as well as allowing for problem tracing and is crucial to facilitating high performance Grid computing.
http://hepunx.rl.ac.uk/edg/wp3/

The EDG Networking work package (WP7) is a component of the European Data Grid project. This project, supported by the European Union, aims at providing distributed computing power in a transparent way for the end user. WP7 is responsible for the provision of network infrastructure and services for the EDG testbeds.
http://www.gridpp.ac.uk/wp7/index.html

The EDG Replica Management Service supports the generation and storage of replica data at multiple globally distributed sites. The reference implementation of this system is called Reptor.
http://edg-wp2.web.cern.ch/edg-wp2/docs/ReplicaManager/ReptorPaper.pdf

The EDG Storage Element (SE) is designed to sit between the client and the Mass Storage System (MSS), to hide the MSS differences from the client and to allow access to the MSS using protocols that it does not naturally support. In addition to this role, the SE will also provide other Grid functions as applied to data access. For example: security; access control; monitoring; logging; network transfer estimation.
http://web01.esc.rl.ac.uk/projects/DataGrid/wp5/

The EDG workload management section of the European Data Grid project has the goal of defining and implementing an architecture for distributed scheduling and resource management in a Grid environment.
http://server11.infn.it/workload-grid/

The twelve EDG work packages cover middleware, infrastructure, applications and management.
http://web.datagrid.cnr.it/servlet/page?_pageid=1429&_dad=portal30&_schema=PORTAL30&_mode=3

Enterprise JavaBeans (EJB) is an example of server-side component technology that is widely used for building business applications. The EJB component model simplifies the development of middleware applications by providing automatic support for services such as transactions, security, database connectivity, and more.
http://java.sun.com/products/ejb/

eSNW (e-Science North-West) is one of the eight Regional Centres in the EPSRC/DTI e-Science Core Technology programme, and supports the development of e-Science activities within Wales and the South-West of England. eSNW specializes in BioMedical e-Science and aspires to be a primary source of top quality curated repositories
http://www.esnw.ac.uk/

The Fault Tolerant CORBA specification is intended to support large mission-critical systems (such as air traffic control and defense systems), smaller mission critical systems (such as medical systems), as well as embedded applications, communications systems, and enterprise applications.
http://www.omg.org/cgi-bin/doc?ptc/2000-04-04

GADS (Grid Access Database System) is used to query and request data from large gridded datasets. Data may be selected from flat files in various formats, and these data subsets are prepared in a user specified format for download or delivery to further web/grid services.
http://ws1.esc.rl.ac.uk/documents/staff/andrew_woolf/woolf03.abstract.htm

GALE (Grid Access Language for HPC Environments) is an HPC workflow vocabulary that uses key Grid services to provide a grid-level scripting language for users and problem-solving environments.
http://vir.sandia.gov/drmweb/docs/gale.pdf

GARA (General-purpose Architecture for Reservation and Allocation) is a comprehensive architecture for providing applications with Quality of Service for different types of resources, such as networks, CPUs, batch job schedulers, disks, and graphics pipelines. GARA can also provide mechanisms to allow both advance reservations and immediate reservations for quality of service, and enables high-performance computing users to conveniently make and use QoS reservations for complex sets of resources. GARA is part of the Globus project.
http://www-fp.mcs.anl.gov/qos/

GASS (Global Access to Secondary Storage) simplifies the porting and running of applications that use file I/O to the Globus environment. Libraries and utilities are provided to eliminate the need to manually login to sites and ftp files, and to install a distributed file system. The APIs are designed to allow reuse of programs that use Unix or standard C I/O with little or no modification.
http://www.globus.org/gass/

The GAT (Grid Application Toolkit) is a set of coordinated, generic and flexible APIs for accessing Grid services from generic application codes, portals, and data managements systems, together with working implementations provided by the tools developed in the *GridLab* project. The GAT is designed in a modular plug-and-play manner, such that tools developed anywhere that conform to the GAT API will be inter-operable. GAT is part of the *GridLab* project.
http://www.gridlab.org/WorkPackages/wp-1/

The GDMP (Grid Data Mirroring Package) client-server software system is a generic file replication tool that replicates files securely and efficiently from one site to another in a Data Grid environment using several *Globus* tools. In addition, it manages replica catalogue entries for file replicas and thus maintains a consistent view of names and locations of replicated files.
http://project-gdmp.web.cern.ch/project-gdmp/

GGF official documents are processed in several steps. GGF documents generally begin within Working Groups and Research Groups. They are then submitted, upon completion, for review by the Grid Forum Steering Group (GFSG), and are published as part of the GGF document series, based on GFSG review and public comment.
http://www.ggf.org/documents/Drafts/default.htm

The GGF AuthZ-WG (Authorization Frameworks and Mechanisms Working Group) is defining a conceptual grid authorization framework for grid developers with the goal to provide a basis for the design of such grid authorization systems.
http://www.gridforum.org/2_SEC/auth.htm

The GGF GCE-RG (Grid Computing Environments Research Group) is aimed at contributing to the coherence and interoperability of frameworks, portals, PSEs, and other Grid-based computing environments by establishing standards that are required to integrate technology implementations and solutions.
http://www.gridforum.org/7_APM/GCE.htm

The GGF GMA-WG (Grid Monitoring Architecture Working Group) is focused on producing a high-level architecture statement of the components and interfaces needed to promote interoperability between heterogeneous monitoring systems on the Grid.

http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

The GGF GPA-WG (Grid Protocol Architecture Working Group) of the GGF seeks to provide a conceptual framework for discussing the interrelationships, completeness, and minimality of the protocol approach to Grid services.
http://www-itg.lbl.gov/GPA/

GIIS (Grid Index Information Service) provides a means of knitting together arbitrary GRIS services to provide a coherent system image that can be explored or searched by grid applications.
http://www.lesc.ic.ac.uk/services/giis.html

The Globus Replica Catalog supports replica management by providing mappings between logical names for files and one or more copies of the files on physical storage systems.
http://www.globus.org/datagrid/replica-catalog.html

Globus Replica Management integrates the Globus Replica Catalog (for keeping track of replicated files) and GridFTP (for moving data) and provides replica management capabilities for data grids.
http://www.globus.org/datagrid/replica-management.html

The GLUE (Grid Laboratory Uniform Environment) schema aims to define a common conceptual data model to be used for grid resources monitoring and discovery. GLUE is part of the *DATATAG* project.
http://www.cnaf.infn.it/~sergio/datatag/glue/index.htm

GPIR (GridPort Information Repository) aggregates and caches grid and portal related data in support of rapid and easy portal development, and thereby provides *GridPort* with its data persistence needs.
http://www.tacc.utexas.edu/grid/gpir/

The GPT (Grid Packaging Tools) are a collection of packaging tools built around an XML-based packaging data format. This format provides a straightforward way to define complex dependency and compatibility relationships between packages.
http://www.nsf-middleware.org/NMIR3/components/gpt.asp

GRAM (Globus Resource Allocation Manager) is a set of service components that simplifies the use of remote systems by providing a single standard interface for requesting and using remote system resources for the execution of jobs. The most common use of GRAM is remote job submission and control. GRAM is part of the Globus project.
http://www-unix.globus.org/developer/program-execution.html

GridAnt uses the Ant framework to develop a simple yet powerful client side workflow system for Grids. GridAnt can be used to map complex client-side workflows, and also as a simplistic client to test the functionality of different Grid services.
http://www-unix.globus.org/cog/projects/gridant/

Gridconfig is a collection of configuration tools that manages the configuration for *NMI* software components. It provides an easy way to generate and regenerate configuration files in native formats, and to ensure configuration consistency.
http://www.nsf-middleware.org/NMIR2/components/gridconfig.asp

GridFTP (Grid File Transfer Protocol) is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. The GridFTP protocol is based on FTP, the highly-popular Internet file transfer protocol. GridFTP is part of the Globus project.
http://www.globus.org/datagrid/gridftp.html

gridMathematica combines the Mathematica technical computing environment with modern computing clusters and grids to solve demanding problems in mathematics, science, engineering, and finance. gridMathematica provides a quick way to set up and run large calculations by offering a high-level

programming language, a collection of fast and reliable mathematical algorithms, and easy-to-use parallel programming constructs.
http://www.wolfram.com/products/gridmathematica/

GridRPC is a remote procedure call API for Grid computing that provides a basic mechanism for implementing a variety of Grid-aware applications and services.
http://www.eece.unm.edu/~apm/docs/APM_GridRPC_0702.pdf

GridSite allows restricted groups of website users to edit the content of the site and administrators to manage access control through unmodified web browsers using grid credentials - for example, certificates from the e-Science Certification Authority.
http://www.gridpp.ac.uk/authz/gridsite/

The GridSphere portal framework provides an open-source portlet based Web portal. GridSphere enables developers to quickly develop and package third-party portlet web applications that can be run and administered within the GridSphere portlet container.
http://www.gridsphere.org/gridsphere/gridsphere

GRIS (Grid Resource Information Service) provides a uniform means of querying resources on a computational grid for their current configuration, capabilities, and status. Such resources include, but are not limited to computation nodes, data, storage systems, scientific instruments, network links, and databases. GRIS is part of the Globus project.
http://www.globus.org/toolkit/information-infrastructure.html and http://www.globus.org/mds/extending-gris.html

Grokster is a freely-available peer-to-peer file sharing program.
http://www.grokster.com/

GSI (Grid Security Infrastructure) enables secure authentication and communication over an open network. GSI provides a number of useful services for Grids, including mutual authentication and single sign-on.
http://www.globus.org/security/

GSIFTP. GSIFTP is a standard Unix FTP (File Transfer Protocol) program modified to use the GSI libraries for authentication. It replaces the normal password authentication with Globus certificate authentication.
http://www.globus.org/datagrid/gridftp.html

GSI-OpenSSH is a modified version of OpenSSH that adds support for *GSI* authentication, providing a single sign-on remote login capability for the Grid. GSI-OpenSSH can be used to login to remote systems and transfer files between systems without entering a password, relying instead on a valid GSI credential for operations requiring authentication.
http://www.nsf-middleware.org/NMIR3/components/gsissh.asp

GT2 (Globus Toolkit 2) is an open source toolkit that includes software services and libraries for resource monitoring, discovery, and management, plus security and file management.
http://www.globus.org/gt2.4/

GT3 (Globus Toolkit 3) aims to deliver an open source implementation of OGSI, several OGSI-compliant services corresponding to familiar GT2 services, and the ability to create new OGSI-compliant services.
http://www.globus.org/toolkit/gt3-factsheet.html

HotPage is a portal that enables researchers to find information about each of the resources in the NPACI computational grid: technical documentation, operational status, load and current usage, queued jobs, etc.
https://hotpage.npaci.edu/about/

HPC-MW (High Performance Computing Middleware) is an infrastructure for the efficient development of optimized and reliable scientific simulation codes.
http://geofem.tokyo.rist.or.jp/presen_common/ACES/3rdACES/abstracts/ACES02_KN_HPCmw.pdf

Ilab (IPG Virtual Laboratory) provides support for the exhaustive process of building large parameter studies – involving editing hundreds of files, creating shell scripts to run computer jobs, saving data entered, and shipping studies off for distributed processing.
http://www.nas.nasa.gov/ILab/

Internet monitoring is important to the efficient use of Grid resources and has been a topic of much investigation.
http://www.slac.stanford.edu/comp/net/wan-mon/netmon.html

Iperf is a tool to measure maximum TCP bandwidth, allowing the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss.
http://dast.nlanr.net/Projects/Iperf/

The IPMP (IP Measurement Protocol) protocol has been designed to allow routers to participate in active measurement of networks without compromising their core packet forwarding role. Special care has been taken to make the impact on the router minimal. The protocol supports simultaneous delay and path determination with a low impact on the network.
http://watt.nlanr.net/AMP/IPMP/

The IPPM (Internet Protocol Perfomance Metrics) working group of the IETF is developing a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services.
http://www.ietf.org/html.charters/ippm-charter.html

The IRS (Internet Reasoning Service) is a Semantic Web Services framework, which allows applications to semantically describe and execute web services. The IRS supports the provision of semantic reasoning services within the context of the Semantic Web.
http://kmi.open.ac.uk/projects/irs/

IT Innovation is a systems engineering, systems integration and consultancy business dedicated to the innovative application of information technology. It is heavily involved in the UK e-Science programme.
http://www.it-innovation.soton.ac.uk/

JavaNIO refers to the new I/O (NIO) APIs introduced in Java 1.4 that provide new features and improved performance in the areas of buffer management, scalable network and file I/O, character-set support, and regular-expression matching.
http://www.javanio.info/ and http://java.sun.com/j2se/1.4.1/docs/guide/nio/

Jetspeed is an open source implementation of an Enterprise Information Portal, using Java and XML. Jetspeed acts as the central hub where information from multiple sources is made available in an easy to use manner.
http://jakarta.apache.org/jetspeed/site/index.html

Jini is a network architecture for the construction of distributed systems. It provides a flexible infrastructure for delivering services in a network and for creating spontaneous interactions between clients and services.
http://www.jini.org/

JIPANG (Jini-based Portal Augmenting Grids) is a portal system and a toolkit that provides uniform access to a variety of Grid systems and is built on top of Jini distributed object technology.
http://matsu-www.is.titech.ac.jp/~suzumura/jipang/

JISGA (JIni-based Service-oriented Grid Architecture) is a lightweight infrastructure for Grid computing based on Jini and XML technologies. JISGA features workflow composition and enactment, and supports the interoperation of Jini services and Web services.
http://www.cs.cf.ac.uk/User/Yan.Huang/GridWF/JISGA.htm

The JMS (Java Message Service) API is an API for accessing enterprise messaging systems. JMS allows applications to asynchronously send and receive critical data and events. JMS supports both message queueing and publish-subscribe styles of messaging.
http://java.sun.com/products/jms/

The Joint Battlespace Infosphere (JBI) is a combat information management system which provides users with specific information required to perform their functional responsibilities during crisis or conflict. The JBI integrates data from a wide variety of sources, aggregates this information, and distributes the information in the appropriate form and level of detail to users at all echelons.
http://www.rl.af.mil/programs/jbi/default.cfm

Joint Vision 2020 is aimed at the transformation of America's Armed Forces to create a force that is dominant across the full spectrum of military operations – persuasive in peace, decisive in war, preeminent in any form of conflict.
http://www.dtic.mil/jointvision/index.html

JXTA (short for juxtapose, as in side by side) is a set of open, generalized peer-to-peer (P2P) protocols that allow any connected device on the network to communicate and collaborate.
http://www.jxta.org/

The JXTA CMS (Content Management System) is a simple JXTA service that builds off of other JXTA services such as the Resolver and Endpoint Messenger to allow files to be shared and searched for within a peergroup.
http://cms.jxta.org/

Kazaa Media Desktop is P2P software that allows users to search for, download, and use audio/music, document, image, playlist, software, and video files.
http://www.kazaa.com/us/index.htm

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography.
http://web.mit.edu/kerberos/www/

Khoros provides an integrated development environment to allow users to rapidly prototype solutions, develop new software, manage complex software configurations, and integrate diverse software programs into a uniform framework.
http://www.khoral.com/khoros/

K-PKI (Kerberos Leveraged PKI) leverages an existing Kerberos infrastructure to provide a lightweight Public Key Infrastructure (PKI).
http://www.citi.umich.edu/projects/kerb_pki/

LCFG (Local ConFiGuration) is a system for automatically installing and managing the configuration of large numbers of Unix systems. It is particularly suitable for sites with very diverse and rapidly changing configurations.
http://www.lcfg.org/

LCFG(ng) is the new generation of Edinburgh University's large scale configuration system LCFG. The European DataGrid uses core LCFG(ng) software from Edinburgh together with adapted versions of some Edinburgh LCFG(ng) components and new Grid-specific components.
http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/documentation/

LDAP (Lightweight Directory Access Protocol) is a specification for a client-server protocol to retrieve and manage directory information. It was originally intended as a means for clients on PCs to access X.500 directories, but can also be used with any other directory system that follows the X.500 data models.
http://www.innosoft.com/ldapworld/

LeSC (London e-Science Centre) is one of the eight Regional Centres in the EPSRC/DTI e-Science Core Technology programme, and supports the development of e-Science activities within London and the South-East through collaborations with applied scientists in the fields of Materials Modelling, Particle Physics, Bioinformatics, Environmental Modelling, and Engineering.
http://www.lesc.ic.ac.uk/

The LHC (Large Hadron Collider) is an accelerator which brings protons and ions into head-on collisions at higher energies than ever achieved before. This will allow scientists to penetrate still further into the structure of matter and recreate the conditions prevailing in the early universe, just after the "Big Bang".
http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/

Platform LSF can intelligently schedule, and guarantee completion of, batch workload across a distributed, virtualized IT environment, and fully utilize all IT resources regardless of operating system, including desktops, servers, supercomputers and mainframes.
http://www.platform.com/products/LSF/

Mathematica seamlessly integrates a numeric and symbolic computational engine, graphics system, programming language, documentation system, and advanced connectivity to other applications.
http://www.wolfram.com/products/mathematica/

Matlab integrates mathematical computing, visualization, and a powerful language to provide a flexible environment for technical computing.
http://www.mathworks.com/products/matlab/

The Maui Silver Metascheduler is an advance reservation metascheduler. Its design allows it to load balance workload across multiple systems in completely independent administrative domains. How much or how little a system participates in this load-sharing is completely up to the local administration.
http://www.supercluster.org/documentation/silver/silveroverview.html

MCAT is a meta-information catalog system implemented as part of the Data Intensive Computing Environment (DICE) with requirements mainly based on the Storage Resource Broker system (*SRB*).
http://www.npaci.edu/DICE/SRB/mcat.html

MDS (Monitoring and Discovery Service) provides the necessary tools to build an LDAP-based information infrastructure for computational grids. MDS uses the LDAP protocol as a uniform means of querying system information from a rich variety of system components, and for optionally constructing a uniform namespace for resource information across a system that may involve many organizations. MDS is part of the Globus project.
http://www.globus.org/mds/

Morpheus is a peer-to-peer program for searching for, downloading, playing and sharing (mainly) audio files.
http://www.morpheus-os.com/ and http://www.musiccity.com/

MpCCI (Mesh-based parallel Code Coupling Interface) is a code coupling interface for multidisciplinary applications. It enables industrial users as well as code owners to combine different simulation tools.
http://www.mpcci.org/

MPICH-G2 is a grid-enabled implementation of the MPI v1.1 standard based on the MPICH library developed at Argonne National Laboratory. Using services from the Globus Toolkit (e.g., job startup,

security) MPICH-G2 allows users to couple multiple machines, potentially of different architectures, to run MPI applications.
http://www.nsf-middleware.org/NMIR3/components/mpichg2.asp

The MQ (formerly MQ Series) family of products enables customers to integrate people, processes, information, and systems throughout an enterprise. The MQ family is part of IBM WebSphere.
http://www-3.ibm.com/software/integration/mqfamily/

The myGrid workflow enactment engine is a workflow orchestration tool for web services. It can handle *WSDL* based web service invocation, even when those *WSDL* definitions include arbitrary complex types. It can interrogate a standard *UDDI* registry, given preferences from the workflow author, to obtain actual services instances to invoke. It supports an XML workflow definition language that is a based on IBM's *WSFL*, including support for specifying control and data flow.
http://mygrid.man.ac.uk/myGrid/web/components/Workflow/

The myGrid notification service is responsible for delivery and persistence of messages between myGrid core and application services. It is an important building block for myGrid enabling asynchronous publish-subscribe communication and negotiation over quality of services.
http://mygrid.man.ac.uk/myGrid/web/components/NotificationService/

MyGrid provenance metadata is a key feature of the myGrid environment. Provenance data can be broken down into two categories: derivation data and annotations. Derivation data provides the answer to questions about what initial data was used for a result, and how was the transformation from initial data to result achieved.
http://mygrid.man.ac.uk/myGrid/web/components/ProvenanceData/

MyProxy is a credential repository for the Grid. Storing users Grid credentials in a MyProxy repository allows them to retrieve a proxy credential whenever and wherever they need one, without worrying about managing private key and certificate files.
http://www.nsf-middleware.org/NMIR3/components/myproxy.asp

MySpace is a virtual directory space of data items, located in databases or files anywhere on the Grid, but which the user can see and manipulate through an Explorer-type interface. AstroGrid has adapted this idea already to cover both data centre cache provision and community-based private storage.
http://wiki.astrogrid.org/bin/view/Astrogrid/MySpace

The NaradaBrokering project aims to provide a unified messaging environment that integrates Grid Services, JMS and JXTA. NaradaBrokering is an event brokering system designed to run on a large network of cooperating broker nodes. Narada stands for iNtegrated Asynchronous Real-time Adaptive Distributed Architecture.
http://www.naradabrokering.org/

NCW (Network Centric Warfare) represents a powerful set of warfighting concepts and associated military capabilities that allow warfighters to take full advantage of all available information and bring all available assets to bear in a rapid and flexible manner.
http://www.c3i.osd.mil/NCW

The NEESGrid Metadata Service provides access to metadata about about a variety of entities, such as experiments, researchers, apparatus, events and facilities. The service is designed to allow remote clients to browse, update, and otherwise manage metadata representing these entities of interest.
http://www.neesgrid.org/repository/MetadataService_v1_0.pdf and http://www.neesgrid.org/repository/

NEReSC (North-East Regional e-Science Centre) is one of the eight Regional Centres in the EPSRC/DTI e-Science Core Technology programme, and supports the development of e-Science activities within the North-East of England. Its area of specialization is data intensive Grid applications.
http://www.neresc.ac.uk/index.html

NERSC (National Energy Research Scientific Computing Center) is one of the largest unclassified scientific supercomputing centers in the world. NERSC provides high-performance computing tools and expertise that enable large-scale computational science, in which large, interdisciplinary teams of scientists attack fundamental problems in science and engineering that require massive calculations and have broad scientific and economic impacts.
http://www.nersc.gov/

NeSC (National e-Science Centre) of the UK EPSRC/DTI e-Science Core Technology programme supports the development of e-Science activities within Scotland. NeSC seeks to stimulate and sustain the development of e-Science in the UK, to contribute significantly to its international development and to ensure that its techniques are rapidly propagated to commerce and industry. NeSC also develops advances in scientific data curation and analysis and to be a primary source of top quality systems and repositories that enable management, sharing and best use of research data.
http://www.nesc.ac.uk

Nimrod is a specialized parametric modeling system. Nimrod uses a simple declarative parametric modeling language to express a parametric experiment and provides machinery that automates the task of formulating, running, monitoring, and collating the results from the multiple individual experiments. Nimrod also incorporates a distributed scheduling component that can manage the scheduling of individual experiments to idle computers in a local area network.
http://www.csse.monash.edu.au/~davida/nimrod.html/

Nimrod/G extends the Nimrod project to Grid environments. It examines the parameterisation of serial programs to create embarrassingly parallel programs, and distributed scheduling to provide a uniform guaranteed completion time.
http://www.csse.monash.edu.au/~sgaric/nimrod/

OGC Web Services (Open GIS Consortium Web Services) are an evolutionary, standards-based framework that enables seamless integration of a variety of online geoprocessing and location services. OGC Web Services will allow distributed geoprocessing systems to communicate with each other using technologies such as XML and HTTP.
http://ip.opengis.org/ows2/

OGSA (Open Grid Service Architecture) is intended to be a broad and comprehensive architecture for the Grid based on *Grid services*. The specification of OGSA is being discussed by the OGSA working group of the Global Grid Forum.
http://www.globus.org/ogsa/

OGSI (Open Grid Service Infrastructure) is the current Grid service specification extending Web services to support dynamically-created *stateful services*.
http://www.gridforum.org/ogsi-wg/

OPeNDAP (Open-source Project for a Network Data Access Protocol) is a protocol that provides a discipline-neutral means of requesting and providing data across the World Wide Web. The goal is to allow end users, whoever they may be, to access immediately whatever data they require in a form they can use, all while using applications they already possess and are familiar with.
http://opendap.org/

The OpenJava MOP (Metaobject Protocol) is the extension interface of the OpenJava language, which is an extensible language based on Java. Through the MOP, programmers can customize the language to implement a new language mechanism.
http://www.csg.is.titech.ac.jp/openjava/

The OWL Web Ontology Language is a semantic markup language for publishing and sharing ontologies on the World Wide Web. OWL is developed as a vocabulary extension of RDF (the Resource Description Framework) and is derived from the DAML+OIL Web Ontology Language.
http://www.w3.org/TR/owl-guide/

Pacman is a package manager that enables a user to transparently fetch, install and manage software packages.
http://physics.bu.edu/~youssef/pacman/ and http://www.lsc-group.phys.uwm.edu/vdt/why_pacman.html

The PACX-MPI (PArallel Computer eXtension) library enables scientists and engineers to seamlessly run MPI-conforming parallel application on a Computational Grid, such as a cluster of high-performance computers, connected through high-speed networks or even the Internet. The parallel application does not have to be changed in any way but only recompiled and linked against PACX-MPI.
http://www.hlrs.de/organization/pds/projects/pacx-mpi/

Pastry is a generic, scalable and efficient substrate for peer-to-peer applications. Pastry nodes form a decentralized, self-organizing and fault-tolerant overlay network within the Internet. Pastry provides efficient request routing, deterministic object location, and load balancing in an application-independent manner. Furthermore, Pastry provides mechanisms that support and facilitate application-specific object replication, caching, and fault recovery.
http://research.microsoft.com/~antr/Pastry/

PBS (Portable Batch System) is a flexible batch queuing and workload management system that operates on networked, multi-platform UNIX environments, including heterogeneous clusters of workstations, supercomputers, and massively parallel systems.
http://www.openpbs.org/main.html

PBS Pro is the professional version of *PBS*. It operates in networked multi-platform UNIX environments, and supports heterogeneous clusters of workstations, supercomputers, and massively parallel systems.
http://www.pbspro.com/

Pegasus is a configurable system that can map and execute complex workflows on the Grid. Pegasus has been integrated with the *GriPhyN Chimera* system. In that configuration, Pegasus receives an abstract workflow description from Chimera, produces a concrete workflow, and submits it to *DAGMan* for execution.
http://www.isi.edu/~deelman/pegasus.htm
A Personal Condor is a version of *Condor* running as a regular user, without any special privileges. The idea is that you can use your Personal Condor to run jobs on your local workstations and have Condor keep track of their progress, and then through "flocking" access the resources of other Condor pools. Additionally, you can "*Glide-in*" to Globus-Managed resources, and create virtual-condor pool by running the Condor daemons on the *Globus* resources, and then letting your Personal Condor manage those resources.
http://www.cs.wisc.edu/condor/condorg/README

Polycom is a commercial videoconferencing company.
http://www.polycom.com/

The PST (Practical Supercomputing Toolkit) intends to mitigate problems associated with non-expert users making use of supercomputing facilities.
http://pstoolkit.org/

PUNCH (Purdue University Network Computing Hubs) is a platform for Internet computing that turns the World Wide Web into a distributed computing portal., so that users can access and run programs via standard Web browsers.
http://punch.purdue.edu/

PVM (Parallel Virtual Machine) is a software package that permits a heterogeneous collection of Unix and/or Windows computers hooked together by a network to be used as a single large parallel computer. Thus large computational problems can be solved more cost effectively by using the aggregate power and memory of many computers.
http://www.csm.ornl.gov/pvm/pvm_home.html

R-GMA (Relational Grid Monitoring Architecture) is part of the European DataGrid project. It is based on the Grid Monitoring Architecture from the Global Grid Forum (GGF). In R-GMA the implementation is based on a relational model. R-GMA makes information from Producers available to Consumers as relations (tables). The R-GMA implementation uses HTTP Servlet technology. Communication with the servlets is achieved via an API.
http://marianne.in2p3.fr/datagrid/documentation/EDG-Users-Guide/node16_mn.html and
http://hepunx.rl.ac.uk/edg/wp3/

The RPM Package Manager is a command line driven package management system capable of installing, uninstalling, verifying, querying, and updating computer software packages.
http://www.rpm.org/

The Scientific Data Mining, Integration and Visualisation workshop considered the challenges for e-sciencists in enabling the effective extraction, integration, analysis and presentation of knowledge from the data avalanche.
http://www.anc.ed.ac.uk/sdmiv/

SCIRun is a scientific problem-solving environment with particular support for computational steering and visualization via a dataflow style of interface.
http://software.sci.utah.edu/scirun.html

The SkyServer portal provides public access to the Sloan Digital Sky Survey.
http://skyserver.sdss.org/en/

SlashGrid is a framework for adding new file systems to Unix systems, especially file systems controlled by Grid credentials and using Grid protocols to give access to remote resources via local, virtual file systems.
http://www.gridpp.ac.uk/authz/slashgrid/

SLP (the IETF Service Location Protocol) is a decentralized, lightweight, scalable and extensible protocol for service discovery within a site.
http://www.ietf.org/html.charters/svrloc-charter.html

SmartFrog (Smart Framework for Object Groups) is a framework for the development of configuration-driven systems. It was originally designed as a framework for building and managing large, distributed monitoring systems where flexible configurations are essential. The framework defines systems and sub-systems as collections of software components with certain properties. It provides mechanisms for describing these component collections, deploying and instantiating them and then managing them during their entire lifecycle.
http://www-uk.hpl.hp.com/smartfrog/

SOAP (Simple Object Access Protocol) is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.
http://www.w3.org/TR/SOAP/

Spitfire is a project of the European Data Grid Project. It provides a Grid enabled middleware service for access to relational databases. Currently it consists of the Spitfire Server module and the Spitfire Client libraries and command line executables.
http://spitfire.web.cern.ch/Spitfire/

SRB (Storage Resource Broker) is a client-server based middle-ware initially developed by San Diego Supercomputer Center in the mid-Nineties to provide uniform access interface to different types of storage devices. SRB provides a uniform API that can be used to connect to heterogeneous resources (filesystems, tape stores, and databases) that may be distributed, and access data sets that may be replicated.
http://www.npaci.edu/dice/srb/ and http://ws1.esc.rl.ac.uk/web/projects/storage_resource_broker

The SRM-WG (Storage Resource Management Working Group) is mainly a forum for the EDG and PPDG groups to discuss storage resource management issues and agree on common software and interfaces.
http://sdm.lbl.gov/srm-wg/

Sun Grid Engine software supports Campus and departmental grids.
http://wwws.sun.com/software/gridware/

SWFL (Service Workflow Language) is an extension of Web Services Flow Language (*WSFL*) for describing applications and higher level services composed of interacting services. SWFL documents describing applications are input to the *JISGA* workflow engine for execution.
http://www.cs.cf.ac.uk/User/Yan.Huang/GridWF/SWFL.htm

The SWOF (Scientific Workspaces of the Future) expedition will create and deploy next-generation collaborative scientific visualization tools and systems for use by distributed communities.
http://www-unix.mcs.anl.gov/fl/research/SWOF/

TOOLSHED is a problem solving environment (PSE) developed by IT Innovation with Bertin of France and ENEL of Italy. The environment uses both STEP and native data formats to manage data exchange and provides CAD import from Numeca, automatic meshing from Numeca and Bertin, computational steering from Sintef and visualisation through the GLView tool from ViewTech.
http://www.it-innovation.soton.ac.uk/services/eng_design/design_cases.shtml#toolshed

UDDI (Universal Description Discovery and Integration) defines a set of services supporting the description and discovery of Web services providers, the Web services they make available, and the technical interfaces that may be used to access those services.
http://uddi.org/ and http://uddi.org/pubs/uddi_v3.htm

UDDI4J is a Java class library that provides an API to interact with a *UDDI* registry.
http://www-124.ibm.com/developerworks/oss/uddi4j/

The UK e-Science Grid Monitoring Service provides an up-to-date snapshot of the UK e-Science Grid and also provides statistical information.
http://rtlin1.dl.ac.uk/gridmon/

United Devices is a company that markets secure Grid solutions. The company's flagship platform, Grid MP enables any organization to coordinate and share existing computing, application, data, storage, and network resources across departmental and geographically dispersed organizations – or to outsource processing needs to a secure private grid.
http://www.ud.com/home.htm

The VDT (Virtual Data Toolkit) is a set of software that supports the needs of the research groups and experiments involved in the *Griphyn* project. It contains fundamental grid software, such as *Condor* and *Globus*, and virtual data software.
http://www.lsc-group.phys.uwm.edu/vdt/

VOMS (Virtual Organization Membership Service) is part of the European DataGrid project. It provides information on the user's relationship with thier Virtual Organization: their groups, roles and capabilities. The service is basically a simple account database, which serves the information in a special format (VOMS credential).
http://edg-wp2.web.cern.ch/edg-wp2/security/voms.html

VRVS (Virtual Room Videoconferencing System) is a web-oriented system for videoconferencing and collaborative work over IP networks.
http://www.vrvs.org/

WebDAV (Web-based Distributed Authoring and Versioning) is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.
http://www.webdav.org/

WebEx is company that markets software for online meetings, web conferencing, and videoconferencing.
http://www.webex.com/

WebSphere is a comprehensive development and deployment environment for building, testing, and deploying on-demand e-business applications. It is an IBM product.
http://www-3.ibm.com/software/webservers/

WeSC (Welsh e-Science Centre) is one of the eight Regional Centres in the EPSRC/DTI e-Science Core Technology programme, and supports the development of e-Science activities within Wales and the South-Wast of England. WeSC specializes in problem-solving environments, the interoperation of heterogeneous databases, and distributed visualization.
http://www.wesc.ac.uk/

The WfMC (Workflow Management Coalition) seeks to increase the value of customers' investment in workflow technology, decrease the risk of using workflow products, and expand the workflow market through increasing awareness for workflow.
http://www.wfmc.org/

The WRG (White Rose Grid) brings together those researchers from the Yorkshire region who are engaged in e-Science activities and through these in the development of Grid technology. The initiative focuses on building, expanding and exploiting the Grid emerging infrastructure, which employs many components to create a collaborative environment for research computing in the region.
http://www.wrgrid.org.uk/

WS-Addressing provides transport-neutral mechanisms to address Web services and messages. Specifically, this specification defines XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in messages. This specification enables messaging systems to support message transmission through networks that include processing nodes such as endpoint managers, firewalls, and gateways in a transport-neutral manner.
http://msdn.microsoft.com/ws/2003/03/ws-addressing/

WS-ReliableMessaging is a protocol that allows messages to be delivered reliably between distributed applications in the presence of software component, system, or network failures. The protocol is described in this specification in an independent manner allowing it to be implemented using different network transport technologies.
http://www-106.ibm.com/developerworks/library/ws-rm/

WSCL (Web Services Conversation Language) allows the abstract interfaces of Web services to be defined. WSCL specifies the XML documents being exchanged, and the allowed sequencing of these document exchanges. WSCL conversation definitions are themselves XML documents and can therefore be interpreted by Web services infrastructures and development tools.
http://www.w3.org/TR/wscl10/

WSDL (Web Services Description Language) is the standard way of specifying the interfaces and attributes of a service through an XML document.
http://www.w3.org/2002/ws/desc/ and http://www.w3.org/TR/wsdl.html

The WSDL4J (Web Services Description Language for Java) toolkit allows the creation, representation, and manipulation of WSDL documents describing services.
http://www-124.ibm.com/developerworks/projects/wsdl4j/

WSFL (Web Services Flow Language) is an XML-based language for describing compositions of Web services.
http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

WSIL (Web Services Inspection Language) is an XML format for assisting in the inspection of a site for available services and a set of rules for how inspection-related information should be made available for consumption.
http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html

WSRP (Web Services for Remote Portals) simplifies integration of remote applications and content into portals. WSRP is a means for content and application providers to provide their services to organizations running portals in an easily-consumed manner.
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp

XPDL is the Workflow Management Coalition's eXtensible Process Definition Language. XPDL provides a means for defining workflow processes in XML.
http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf

15.5 References

The enigmatic symbol **(↑)** at its end indicates that this reference is described above in one of the three subsections on National and Regional Programmes, Projects, Systems and Tools.

1) [AccessGrid]**(↑)** Access Grid Audio-video conferencing http://www.accessgrid.org/ .
2) [Agrawal03A] Sudesh Agrawal, Jack Dongarra, Keith Seymour, and Sathish Vadhiyar, *NetSolve: Past, Present, and Future; A Look at a Grid Enabled Server*, Chapter 24 of [Berman03A]
3) [Akenti]**(↑)** Akenti authorization system from Lawrence Berkeley Laboratory http://www-itg.lbl.gov/Akenti/
4) [AKT]**(↑)** AKT: Advanced Knowledge Technologies Project http://www.aktors.org
5) [ALICELHC]**(↑)** ALICE: A Large Ion Collider Experiment at CERN LHC http://www.cern.ch/ALICE/
6) [AlienGrid]**(↑)** Alien Grid developed for CERN LHC ALICE [ALICELHC] experiment http://alien.cern.ch/
7) [Allen03A] Gabrielle Allen, Tom Goodale, Michael Russell, Edward Seidel and John Shalf, *Classifying and Enabling Grid Applications*, Chapter 23 of [Berman03A]
8) [AMPNLANR] Active Measurement Project (AMP) Homepage, http://watt.nlanr.net//active/intro.html
9) [AstroGrid]**(↑)** AstroGrid: UK Virtual Observatory http://www.astrogrid.org/
10) [AVS]**(↑)** AVS Advanced Visual Systems http://www.avs.com/
11) [Axis]**(↑)** Axis: Apache Web Service Toolkit http://ws.apache.org/axis/
12) [Berman03A] *Grid Computing: Making the Global Infrastructure a Reality* edited by Fran Berman, Geoffrey Fox and Tony Hey, John Wiley & Sons, Chichester, England, ISBN 0-470-85319-0, March 2003. http://www.grid2002.org
13) [Berman03B] Fran Berman Geoffrey Fox Tony Hey, *The Grid: Past, Present, Future*, Chapter 1 of [Berman03A]
14) [BiologyWB]**(↑)** Biology Work Bench at SDSC, http://workbench.sdsc.edu/

15) [Bivens01A](↑) H. Bivens and J. Beiriger, *GALE: Grid Access Language for HPC Environments*. http://vir.sandia.gov/drmweb/docs/gale.pdf. SAND 2001-0860A Technical report.
16) [BPEL4WS](↑) BPEL4WS: F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana, *BPEL4WS, Business Process Execution Language for Web Services*, Version 1.0. Available from http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/.
17) [BPWS4J](↑) BPWS4J: IBM's implementation of BPEL4WS http://www.alphaworks.ibm.com/tech/bpws4j
18) [Brooke03] John Brooke and Donald Fellows, *Abstraction of functions for resource brokers*, GGF-GPA discussion document, Feb. 17 2003, http://www.gridforum.org/Meetings/ggf7/docs/default.htm
19) [Brownlee97A] Brownlee, N. Mills, C. and G. Ruth, *Traffic Flow Measurement: Architecture*, January 1997, IETF RFC 2063. http://www.faqs.org/rfcs/rfc2063.html
20) [Bruneton00A] Bruneton, E., Riveill, M., *JavaPod: an Adaptable and Extensible Component Platform*, Proc. Reflective Middleware 2000, http://www.comp.lancs.ac.uk/computing/rm2000/paper-list.htm.
21) [Buyya02A] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger, *Economics Paradigm for Resource Management and Scheduling in Grid Computing*, Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-15, pages 1507-1542, 2002.
22) [Cactus](↑) Cactus Grid Computational Toolkit http://www.cactuscode.org
23) [CapitalRadio](↑) Capital Radio plc UK http://www.capitalradiogroup.com/
24) [Casanova03A] Henri Casanova and Fran Berman, *Parameter Sweeps on the Grid with APST*, Chapter 33 of [Berman03A]
25) [CCA](↑) CCA: Common Component Architecture Forum http://www.cca-forum.org/
26) [CCLRCeSCDP-A](↑) CCLRC e-Science Centre Data Portal http://ws1.esc.rl.ac.uk/web/projects/dataportal
27) [CCLRCeSCDP-B](↑) CCLRC e-Science Centre Data Portal Trial access http://esc.dl.ac.uk:9000/dataportal/index.html
28) [CCLRCeSCHPC](↑) CCLRC e-Science Centre Data HPC Portal http://esc.dl.ac.uk/HPCPortal/
29) [CCSDS-A] Consultative Committee for Space Data Systems (CCSDS) SLE or Space Link Extension standard http://wwwclassic.ccsds.org/all_books.html
30) [CEOS](↑) CEOS: Committee on Earth Observation Satellites, http://www.ceos.org/
31) [CERN](↑) CERN: Particle Physics Laboratory in Geneva, http://public.web.cern.ch/public/
32) [CGLIndiana] Community Grids Laboratory at Indiana University http://grids.ucs.indiana.edu/ptliupages/
33) [CHEFportal](↑) CHEF: CompreHensive collaborativE Framework from the University of Michigan http://www.chefproject.org/
34) [Chien03A] Andrew A. Chien, *Architecture of an Commercial Enterprise Desktop Grid: The Entropia System*, Chapter 11 of [Berman03A]
35) [Chimera](↑) Chimera Virtual Data System from GryPhyn http://www.griphyn.org/chimera/
36) [Chivers03A] Howard Chivers *Grid Security: Problems and Potential Solutions*. University of York, UK, Department of Computer Science, Yellow Report YCS-2003-354 available at http://www.cs.york.ac.uk/ftpdir/reports/
37) [CiscoIOS] Cisco IOS NetFlow, http://www.cisco.com/warp/public/732/Tech/nmp/index.shtml
38) [Claffy95A] Claffy, K., Braun, H. and G. Polyzos, *A parameterizable methodology for Internet traffic flow profiling*, IEEE JSAC, Special Issue on the Global Internet, 1995
39) [Clarke01A] Clarke, M., Blair, G.S., Coulson, G., *An Efficient Component Model for the Construction of Adaptive Middleware*, IFIP/ACM Middleware 2001, Heidelberg, Nov 2001.
40) [CoABS-A](↑) CoABS Grid *http://coabs.globalinfotek.com* from [CoABS-B]
41) [CoABS-B](↑) Darpa Control of Agent-based Systems CoABS program http://www.darpa.mil/ipto/research/coabs/
42) [CoAKTinG](↑) CoAKTinG (Collaborative Advanced Knowledge Technologies in the Grid) http://www.aktors.org/coakting/
43) [CoaxGrid](↑) Coalition Agents Experiment http://www.aiai.ed.ac.uk/project/coax
44) [CombeChem](↑) Comb-e-Chem Project http://www.combechem.org/.
45) [Condor-ClassAds](↑) Condor ClassAds Classified Advertisements to match computers and jobs http://www.cs.wisc.edu/condor/classad/
46) [Condor](↑) Condor Home Page http://www.cs.wisc.edu/condor/condorg/ see also [Thain03A]

47) [CondorG] Condor-G allowing Globus to access Condor pools http://www.nsf-middleware.org/NMIR3/components/condorg.asp
48) [CondorGlideIn](↑) Condor Glide-in allowing Globus resources to be used in a Condor Pool http://www.cs.wisc.edu/condor/manual/v6.4/condor_glidein.html
49) [CORBANotification](↑) CORBA Junction: CORBA 3.0 Notification Service http://www-106.ibm.com/developerworks/webservices/library/co-cjct8/?dwzone=webservices
50) [COVISA] COVISA: Collaborative Visualization & Scientific Analysis at Leeds University http://www.comp.leeds.ac.uk/vis/covisa/covisa.html
51) [COVISAG] COVISA-G: Globus-enabled Collaborative Visualization demonstration http://www.visualization.leeds.ac.uk/CovisaG/
52) [Curation-A](↑) Seminar sponsored by the Digital Preservation Coalition and the British National Space Centre, Digital Curation: digital archives, libraries, and e-science, London, 19 October 2001, http://www.dpconline.org/graphics/events/digitalarchives.html (Several presentations available from this link)
53) [DAGMan](↑) DAGMan or Directed Acyclic Graph Manager from the Condor project. Available from http://www.cs.wisc.edu/condor/dagman/. See also [Thain03A]
54) [DAME](↑) DAME Distributed Aircraft Maintenance Environment project http://www.cs.york.ac.uk/dame/
55) [DataTAG](↑) DataTAG European Grid Network Infrastructure http://datatag.web.cern.ch/datatag
56) [DeRoure01A] D. De Roure, N. Jennings, and N. Shadbolt. *Research Agenda for the Semantic Grid: A Future e-Science Infrastructure*. Technical report UKeS-2002-02, UK e-Science Technical Report Series, National e-Science Centre, Edinburgh, UK. December 2001. See Chapter 17 of [Berman03A].
57) [DIAMANT](↑) DIAMANT: Digital Film Restoration Environment http://www.it-innovation.soton.ac.uk/research/grid/diamant.shtml
58) [DiscoveryNet-A](↑) DiscoveryNet Project http://www.discovery-on-the-net/
59) [DiscoveryNet-B] DiscoveryNet Workflow as described in section A.7.2.3 of [GapAnalysis]
60) [Dongarra02A] *The Sourcebook of Parallel Computing* edited by Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, and Andy White, Morgan Kaufmann, November 2002
61) [Dongarra02B] Jack Dongarra, Geoffrey Fox, John Rice *Problem Solving Environments*, Chapter 14 in [Dongarra02A]
62) [Dowling02A] Dowling, J, Cahill, V., *Dynamic Software Evolution and The K-Component Model*, Distributed Systems Group, Department of Computer Science, Trinity College Dublin.
63) [Dowling03A] Dowling, J., Cahill, V., *The K-Component Architecture Meta-Model for Self-Adaptive Software*, http://www.cs.tcd.ie/publications/tech-reports/reports.01/TCD-CS-2001-50.pdf, 2003.
64) [DPML](↑) DPML: Discovery Process Markup Language described in [Syed02A]
65) [EAVO](↑) EAVO: European Astrophysical Virtual Observatory http://www.euro-vo.org/
66) [EBI](↑) EBI: European Bioinformatics Institute http://www.ebi.ac.uk/
67) [ECMA.NET] .NET standardisation activity at ECMA http://www.dotnetexperts.com/ecma/.
68) [EconomyGrid](↑) Economy Grid, University of Melbourne Australia, http://www.cs.mu.oz.au/~raj/grids/ecogrid/
69) [EDG-A](↑) European DataGrid EDG http://eu-datagrid.web.cern.ch/eu-datagrid/
70) [EDG-B] EDG Software Release Status https://edms.cern.ch/document/333297
71) [EDG-C](↑) EDG: European DataGrid http://www.eu-datagrid.org/
72) [EDGWP1-B](↑) European DataGrid Work Package 1: Workload Management http://server11.infn.it/workload-grid/
73) [EDGWP10] European DataGrid Work Package 10, Biology Applications, http://edg-wp10.healthgrid.org/
74) [EDGWP1Arch] EDG Work Package 1, *Definition of architecture, technical plan and evaluation criteria for scheduling, resource management, security and job description*, 14 Sept. 2001, http://server11.infn.it/workload-grid/docs/DataGrid-01-D1.2-0112-0-3.doc and other technical documents in [EDGWP1-B]
75) [EDGWP2-B] Bibliography of Replica Management related technologies http://edg-wp2.web.cern.ch/edg-wp2/readings.html

76) [EDGWP2RLS] EDG 2.0 RLS Replica Location Service, ROS Replica Optimization Service (using Optor), RSS Replica Subscription service and RMC Replica Management Catalog http://edg-wp2.web.cern.ch/edg-wp2/replication/index.html
77) [EDGWP2RMS](↑) EDG Replica Management Service described in Leanne Guy Peter Kunszt Erwin Laure, Heinz Stockinger Kurt Stockinger, *Replica Management in Data Grids*, http://edg-wp2.web.cern.ch/edg-wp2/docs/ReplicaManager/ReptorPaper.pdf
78) [EDGWP2VOMS](↑) VOMS: Virtual Organization Membership Service from Work Package 2 of European Data Grid http://edg-wp2.web.cern.ch/edg-wp2/security/voms.html
79) [EDGWP3IMS](↑) European DataGrid Work Package 3 Information and Monitoring Services http://hepunx.rl.ac.uk/edg/wp3/
80) [EDGWP3RGMA](↑) European DataGrid EDG WP3 R-GMA Relational Grid Monitoring Architecture http://hepunx.rl.ac.uk/edg/wp3/
81) [EDGWP4LCFG](↑) Next Generation LCFG(ng) From European DataGrid Work Package 4 http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/
82) [EDGWP5](↑) European DataGrid Work Package 5 http://web01.esc.rl.ac.uk/projects/DataGrid/wp5/ with EDG Storage Element including SRM support
83) [EDGWP7](↑) European DataGrid Work Package 7 on Network Services, http://www.gridpp.ac.uk/wp7/index.html
84) [EDGWP7Secdes] EDG Security Design https://edms.cern.ch/file/344562/2.0/DataGrid-07-D7.6-0112-2-0-SecurityDesign.pdf from EDG Work Package 7 [EDGWP7]
85) [EDGWP7Secreq] EDG Security Requirements https://edms.cern.ch/file/340234/4.0/DataGrid-07-D7.5-0111-4-0-SecurityReq.pdf from EDG Work Package 7 [EDGWP7]
86) [EDGWP8HEP] European DataGrid Work Package 8, ASSESSMENT OF THE TESTBED BY HEP APPLICATIONS DURING PROJECT YEAR 2 https://edms.cern.ch/document/375586
87) [EDGWP8HEPCAL](↑) *High Energy Physics Common application Layer HEPCAL* https://edms.cern.ch/file/375586/1.3/HEPCAL_finalreport.doc
88) [EDGWP9] European DataGrid Work Package 9, Earth Observation Applications, http://styx.esrin.esa.it/grid/intro.html
89) [EDGWP](↑) EDG Work Packages http://web.datagrid.cnr.it/servlet/page?_pageid=1429&_dad=portal30&_schema=PORTAL30&_mode=3
90) [eDIKT](↑) eDIKT: e-Science Data Information & Knowledge Transformation at Edinburgh http://www.edikt.org
91) [EJB](↑) EJB: Enterprise JavaBeans component technology http://java.sun.com/products/ejb/
92) [eMaterials](↑) eMaterials: e-Science simulation of complex materials http://www.e-science.clrc.ac.uk/web/projects/complexmaterials
93) [eMinerals](↑) eMinerals: Environment from the molecular level: An e-science project for modelling the atomistic processes involved in environmental issues http://eminerals.org/
94) [Erwin02A] Dietmar W. Erwin, *UNICORE - A Grid Computing Environment*, Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-15, pages 1395-1410, 2002.
95) [ESG](↑) The Earth System Grid http://www.earthsystemgrid.org/
96) [ESNW](↑) e-Science North West Centre (ESNW) http://www.esnw.ac.uk/
97) [EuroGrid](↑) EuroGrid Application Testbed for European GRID computing http://www.eurogrid.org/
98) [ExpSensorGrid] Expeditionary Sensor Grid http://www.nwdc.navy.mil/OperationsHome/CNAN.asp
99) [ExtremeIndiana] Extreme! Computing at Indiana University http://www.extreme.indiana.edu/
100) [Fassino02A] Fassino, J.-P., Stefani, J.-B., Lawall, J., Muller, G., *THINK: A Software Framework for Component-based Operating System Kernels*, Usenix Annual Technical Conference, Monterey (USA), June 10th-15th, 2002.
101) [Fasttrack] Fasttrack P2P Technology http://www.fasttrack.nu
102) [Fensel00A] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein *OIL in a Nutshell*, pp. 1-16 of *Knowledge engineering and knowledge management* : 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2-6, 2000 : proceedings, Rose Dieng, Olivier Corby (eds.), Lecture Notes in Artificial Intelligence, no. 1937, Springer-Verlag, Berlin, 2000. http://link.springer.de/link/service/series/0558/tocs/t1937.htm
103) [FleetGrid] Fleet Battle Experiments http://www.nwdc.navy.mil/products/fbe/default.cfm

104) [Floyd99A] Floyd, S., Fall, K., *Promoting the Use of End-to-End Congestion Control in the Internet*, IEEE/ACM Transaction on Networking, May 1999
105) [Foster01A] Foster, I., Kesselman, C. and Tuecke, S. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. International Journal of High Performance Computing Applications, 15 (3). 200-222. 2001. www.globus.org/research/papers/anatomy.pdf. Chapter 6 of [Berman03A]
106) [Foster03A] Ian Foster, Carl Kesselman Jeffrey M. Nick and Steven Tuecke, *The Physiology of the Grid*, Chapter 8 of [Berman03A]
107) [Foster99A] Foster, I. and Kesselman, C. (eds.). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
108) [Fox03A] Geoffrey Fox, Dennis Gannon and Mary Thomas, *Overview of Grid Computing Environments*, Chapter 20 of [Berman03A]
109) [Fox03B] G. Fox, D. Gannon, M. Pierce, M. Thomas, *Overview of Grid Computing Environments*, Global Grid Forum Informational Document http://www.gridforum.org/documents/
110) [Fox03C] Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko, Sangmi Lee, Shrideep Pallickara, Marlon Pierce, Xiaohong Qiu, Xi Rao, Ahmet Uyar, Minjun Wang, Wenjun Wu, *Peer-to-Peer Grids*, Chapter 18 of [Berman03A]
111) [Fox03D] Geoffrey Fox and Marlon Pierce, *Grid Computing Environments Shells*, http://grids.ucs.indiana.edu/ptliupages/publications/GGFWPGCEShell.doc
112) [FTCORBA](↑) Fault Tolerant CORBA http://www.omg.org/cgi-bin/doc?ptc/2000-04-04
113) [FTShell] Fault tolerant Shell v2.0 released with the Virtual Data Toolkit VDT 1.1.8 http://www.lsc-group.phys.uwm.edu/vdt/contents.html
114) [GADS](↑) GADS: Grid Access Data Service for Environmental data in Andrew Woolf, Keith Haines and Chunlei Liu, *A Web Service Model for Climate Data Access on the Grid*, http://ws1.esc.rl.ac.uk/documents/staff/andrew_woolf/woolf03.abstract.htm
115) [Gannon03A] Dennis Gannon, Rachana Ananthakrishnan, Sriram Krishnan, Madhusudhan Govindaraju, Lavanya Ramakrishnan, Aleksander Slominski, *Grid Web Services and Application Factories*, Chapter 9 of [Berman03A]
116) [GAPtk] GAPtk: Grid Applications Portals toolkit of generic advanced visualisation utilities based on Web and Grid services http://ws1.esc.rl.ac.uk/web/projects/gaptk
117) [GAT](↑) GAT: Grid Application Toolkit from GridLab http://www.gridlab.org/WorkPackages/wp-1/
118) [GapAnalysis] Geoffrey Fox, David Walker, *e-Science Gap Analysis*, June 30 2003
119) [Gateway](↑) Gateway Computational Portal using Kerberos Security http://www.gatewayportal.org from Community Grids Laboratory at Indiana University
120) [GDMP](↑) GDMP: Grid Data Mirroring Package from EDG and PPDG [PPDG] with Globus-enabled file replication http://project-gdmp.web.cern.ch/project-gdmp/
121) [GEMSS](↑) GEMSS: Grid-Enabled Medical Simulation Services http://www.ccrl-nece.de/gemss/
122) [GEODISE](↑) GEODISE e-Science project in Engineering design and optimization http://www.geodise.org/
123) [Georgatos01A] Georgatos, F., Gruber, F., Karrenberg, D., Santcroos, M., Susanj, A., Uijterwaal, H., Wilhelm, R., *Providing Active Measurements as a Regular Service for ISP's*, Passive and Active Measurement Workshop (PAM2001), Amsterdam, NL, April 23-24, 2001
124) [GGF-A](↑) Global Grid Forum http://www.gridforum.org
125) [GGF-B](↑) Global Grid Forum "official" documents http://www.ggf.org/documents/Drafts/default.htm
126) [GGFAuth](↑) Global Grid Forum Authorization Frameworks and Mechanisms Working Group (AuthZ-WG) http://www.gridforum.org/2_SEC/auth.htm
127) [GGFGCE-A](↑) Grid Computing Environments GCE Research Group of the Global Grid Forum, http://www.gridforum.org/7_APM/GCE.htm.
128) [GGFGCE-B] Grid Computing Environments Research Group Meetings http://www.computingportals.org/meetings/ggf_sum.php. The links to GGF5 and GGF6 both have meetings with workflow discussions.
129) [GGFGCE-C](↑) Grid Portal Architecture Workshop March 6 2003, GGF7 Tokyo, http://www.computingportals.org/meetings/ggf7/
130) [GGFGESA] GESA-WG: Grid Economic Services Architecture http://www.gridforum.org/3_SRM/gesa.htm

131) [GGFGPA](↑) Grid Protocol Architecture Working Group of Global Grid Forum http://www-itg.lbl.gov/GPA
132) [GGFGPACoreGrid](↑) Core Grid Functions from the GPA (Grid Protocol Architecture Working Group of GGF) http://www.gridforum.org/Meetings/ggf7/drafts/CoreGridFunctions.v3.1.Word95.doc
133) [GGFRUS] RUS-WG: OGSA Resource Usage Service http://www.gridforum.org/3_SRM/rus.htm
134) [Globus-A](↑) Globus Project http://www.globus.org
135) [Globus-B](↑) GT2 Globus Toolkit http://www.globus.org/gt2.4/download.html
136) [Globus-C](↑) GT3: Globus Toolkit 3 http://www.globus.org/toolkit/gt3-factsheet.html
137) [GlobusCAS](↑) CAS: Community Authorization Service http://www.globus.org/security/CAS/
138) [GlobusGASS](↑) GASS: Global Access to Secondary Storage http://www.globus.org/gass/
139) [GlobusGIIS](↑) Globus Index Information Service (hierarchical LDAP-based directory service) http://www.lesc.ic.ac.uk/services/giis.html
140) [GlobusGRAM](↑) GRAM Globus Resource Allocation Manager http://www-unix.globus.org/api/c-globus-2.2/globus_gram_documentation/html/index.html
141) [GlobusGridFTP](↑) GridFTP: high-performance, secure, reliable data transfer protocol http://www.globus.org/datagrid/gridftp.html
142) [GlobusGRIS](↑) GRIS: Grid Resource Information Service http://www.globus.org/mds/extending-gris.html
143) [GlobusMDS](↑) MDS: Globus Monitoring and Discovery Service http://www.globus.org/mds/
144) [GlobusPegasus](↑) Globus Pegasus Planning System in Data Management: The Globus Perspective, Globus World January 2003, http://www.globusworld.org/globusworld_web/track2/4_DataManagement1.pdf
145) [GlobusReplCat](↑) Globus Replica Catalog http://www.globus.org/datagrid/replica-catalog.html
146) [GlobusReplMan](↑) Globus Replica Management http://www.globus.org/datagrid/replica-management.html
147) [GLUESchema](↑) Grid Laboratory Uniform Environment (GLUE) schema from European (DataTAG) and US particle physics Grid collaboration [DataTAG] [Trillium]: http://www.cnaf.infn.it/~sergio/datatag/glue/index.htm
148) [Godiva](↑) GODIVA: Grid for Ocean Diagnostics, Interactive Visualisation and Analysis, http://umbriel.dcs.gla.ac.uk/NeSC/action/projects/project_action.cfm?title=81
149) [GommansWS] Leon Gommans, University of Amsterdam, http://carol.wins.uva.nl/~lgommans/
150) [GPIR](↑) GPIR GridPort Information Repository http://www.tacc.utexas.edu/grid/gpir/
151) [GPT](↑) GPT: XML based Grid Packaging Tools for NSF Middleware Initiative http://www.nsf-middleware.org/NMIR3/components/gpt.asp
152) [GrADS](↑) GrADS (Grid Application Development Software Project http://www.hipersoft.rice.edu/grads/
153) [GRIA](↑) GRIA: Grid Resources for Industrial Applications http://www.gria.org/
154) [GridANT](↑) GridAnt workflow system http://www-unix.globus.org/cog/projects/gridant/
155) [GridBus](↑) Gridbus technologies for utility computing, University of Melbourne Australia, http://www.cs.mu.oz.au/~raj/grids/
156) [Gridconfig](↑) Gridconfig collection of tools to manage the configuration for NMI software components http://www.nsf-middleware.org/NMIR2/components/gridconfig.asp
157) [GridLab](↑) GridLab: A Grid Application Toolkit and Testbed, http://www.gridlab.org/
158) [GridPP](↑) GridPP: The Grid for UK Particle Physics, http://www.gridpp.ac.uk/
159) [GridRPC](↑) GridRPC Grid Remote Procedure Call http://www.eece.unm.edu/~apm/docs/APM_GridRPC_0702.pdf
160) [GridSite](↑) GridSite Web Site management from GridPP http://www.gridpp.ac.uk/authz/gridsite/
161) [GridSphere](↑) GridSphere portlet based Web portal development framework Project from GridLab project [GridLab] (Albert-Einstein-Institute, Max-Planck-Institute in Potsdam) http://www.gridsphere.org/gridsphere/gridsphere
162) [GRIDSTART](↑) GRIDSTART: EU Project integrating the other (around 20) European Union Grid Projects, http://www.gridstart.org/
163) [GridSystems](↑) Grid Systems S.A. http://www.gridsystems.com
164) [GridWeaver](↑) GridWeaver UK e-Science fabric management project http://www.epcc.ed.ac.uk/gridweaver/

165) [Grimshaw03A] Andrew S. Grimshaw, Anand Natrajan, Marty A. Humphrey, Michael J. Lewis, Anh Nguyen-Tuong, John F. Karpovich, Mark M. Morgan, Adam J. Ferrari, *From Legion to Avaki: The Persistence of Vision*, Chapter 10 of [Berman03A]
166) [GRIP](↑) GRIP Grid Interoperability Project to integrate Globus and Unicore http://www.grid-interoperability.org/
167) [GriPhyn](↑) GriPhyn Grid Physics Network http://www.griphyn.org/index.php
168) [Grokster](↑) Grokster peer-to-peer File Sharing System http://www.grokster.com/
169) [Groove](↑) Groove Networks Collaboration Technology http://www.groove.net/
170) [GSIOpenSSH](↑) GSI-OpenSSH version of OpenSSH with support for GSI authentication http://www.nsf-middleware.org/NMIR3/components/gsissh.asp
171) [gViz] gViz: Visualization Middleware for e-Science, http://www.visualization.leeds.ac.uk/gViz/
172) [Haupt03A] Tom Haupt and Marlon Pierce, *Distributed object-based grid computing environments*, Chapter 30 of [Berman03A]
173) [HDF] HDF: NCSA Hierarchical Data Format http://hdf.ncsa.uiuc.edu/
174) [Hegering98A] Hegering, H-G., Abeck, S., Neumair, B., *Integrated Management of Networked Systems*, Morgan Kaufmann, 1998
175) [Hendler00A] J. Hendler and I. Horrocks, *The DARPA Agent Markup Language*, IEEE Intelligent Systems, vol. 15, no. 6, Nov./Dec. 2000, pp. 69-72.
176) [Horn01A] Paul Horn, IBM, 10/15/2001 presentation at the AGENDA 2001 conference in Scottsdale, AZ, *Autonomic Computing : IBM's Perspective on the State of Information Technology*, http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf
177) [Hoschek03-A] Wolfgang Hoschek, *Peer-to-Peer Grid Databases for Web Service Discovery*, Chapter 19 of [Berman03A]
178) [HotPage](↑) NPACI HotPage http://hotpage.npaci.edu/; see also GridPort [Thomas03A]
179) [HPC-MW](↑) Kengo Nakajima and Hiroshi Okuda, HPC-MW High performance Computing Middleware, http://geofem.tokyo.rist.or.jp/presen_common/ACES/3rdACES/abstracts/ACES02_KN_HPCmw.pdf
180) [HPC-VAO](↑) HPC-VAO environment for vibro-acoustic optimization, http://www.it-innovation.soton.ac.uk/research/grid/hpcvao.shtml
181) [Huang03] Yan Huang, *The Role of Jini in a Service-Oriented Architecture for Grid Computing,* PhD thesis, Cardiff University, June 2003.
182) [ICENI](↑) ICENI project of the UK e-Science Program. http://www.lesc.ic.ac.uk/iceni/
183) [IETFICMP] ICMP: IETF RFC 792 Internet Control Message Protocol http://www.ietf.org/rfc/rfc0792.txt
184) [Internet2e2epi](↑) Internet 2 End-to-end performance initiative http://e2epi.internet2.edu/
185) [Iperf](↑) Iperf tool to measure maximum TCP bandwidth http://dast.nlanr.net/Projects/Iperf/
186) [IPG](↑) NASA Information Power Grid http://www.ipg.nasa.gov/
187) [IPMP] IPMP: NLANR's IP Measurement Protocol http://amp.nlanr.net/AMP/IPMP/
188) [IPPM](↑) IETF IP Performance Metrics IPPM http://www.ietf.org/html.charters/ippm-charter.html
189) [IRS](↑) IRS: Internet Reasoning Services http://kmi.open.ac.uk/projects/irs/
190) [ITInnovation](↑) IT Innovation at Southampton http://www.it-innovation.soton.ac.uk/
191) [iVDGL](↑) iVDGL - International Virtual Data Grid Laboratory, http://www.ivdgl.org/index.php
192) [iVOA](↑) International Virtual Observatory Alliance http://www.ivoa.net/
193) [Jain86A] Jain, Raj & Shawn A. Routheir, *Packet Trains - Measurements and a New Model for Computer Network Traffic*, IEEE Journal on Selected Areas in Communications, Vol.4, No.6, September 1986, pp. 986-995.
194) [JavaNIO](↑) "New I/O" for Java. See Ron Hitchens, Java NIO, O'Reilly August 2002, 0-596-00288-2 http://www.javanio.info/
195) [JBIGrid](↑) Joint Battlespace Infosphere http://www.rl.af.mil/programs/jbi/default.cfm
196) [Jetspeed](↑) Apache Jetspeed Portal http://jakarta.apache.org/jetspeed/site/index.html
197) [Jini](↑) Jini Network Technology http://www.jini.org/
198) [JISGA](↑) JISGA: Jini-based Service-oriented Grid Architecture http://www.cs.cf.ac.uk/User/Yan.Huang/GridWF/JISGA.htm
199) [JMS](↑) JMS: Java Message Service http://java.sun.com/products/jms/

200) [Johnston03A] Bill Johnston NASA IPG, DoE Science Grid, *Implementing Production Grids*, Chapter 5 of [Berman03A]
201) [Johnston03B](↑) William E. Johnston et al., *DoE Science Grid*, http://doecollaboratory.pnl.gov/research2/doesciencegrid/2pager_march2003.pdf
202) [Jones03A] Chris Jones, CERN, *The Future of Grids and e-Science*, http://www.bc2.ch/2003/abstracts_list.html#Jones
203) [JV2020](↑) Joint Vision 2020 http://www.dtic.mil/jointvision
204) [JXTA](↑) Project JXTA Peer-to-peer system http://www.jxta.org/
205) [JXTACMS](↑) JXTA Content Management System http://cms.jxta.org/
206) [Kazaa](↑) Kazaa Media Desktop http://www.kazaa.com/us/index.php
207) [Khoros](↑) Khoros Integrated Development Environment and visual programming environment http://www.khoral.com/
208) [Kohler99A] Kohler, E., Morris, R., Chen, B., Jannotti, J., Kaashoek, M.F., *The Click Modular Router*, ACM SOSP 1999, pp217-231, Dec 1999.
209) [Krishnan01A] Sriram Krishnan, Randall Bramley, Dennis Gannon, Madhusudhan Govindaraju, Rahul Indurkar, Aleksander Slominski, Benjamin Temko, Jay Alameda, Richard Alkire, Timothy Drews, Eric Webb, *The XCAT Science Portal*, http://www.sc2001.org/papers/pap.pap287.pdf
210) [Krishnan03A] S. Krishnan, P.Wagstrom, and G. von Laszewski, *GSFL: A Workflow Language for Grid Services*, Submitted to IEEE Internet Computing Special Issue on Grid Computing, July/August 2003. Available from http://patrick.wagstrom.net/research/gsfl.pdf.
211) [Kunszt03A] Peter Z. Kunszt and Leanne P. Guy, The Open Grid Service Architecture and Data Grids, Chapter 15 of [Berman03A]
212) [Laszewski02A] Gregor von Laszewski, Mei-Hui Su, Ian Foster, Carl Kesselman, Quasi Real-Time Microtomography Experiments at Photon Sources, in [Dongarra02A]
213) [Laszewski03A] Gregor von Laszewski, Jarek Gawor, Sriram Krishnan, and Keith Jackson, *Commodity Grid Kits - Middleware for Building Grid Computing Environments*, Chapter 26 of [Berman03A]
214) [LCFG](↑) LCFG Fabric Management System from the University of Edinburgh, http://www.lcfg.org/
215) [LCG](↑) LCG: LHC Computing Grid, http://lcg.web.cern.ch/LCG/
216) [Lee01A] C. Lee, S. Matsuoka, D. Talia, A. Sussman, M. Mueller, G. Allen, J. Saltz, *A Grid Programming Primer*, Advanced Programming Models Research Group of GGF Informational Document, August 2001, http://www.eece.unm.edu/~apm/docs/APM_Primer_0801.pdf
217) [LeSC](↑) LeSC: The London e-Science Centre, http://www.lesc.ic.ac.uk/
218) [LHC](↑) LHC: The Large Hadron Collider at CERN, http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/
219) [Liberty](↑) Liberty Alliance http://www.projectliberty.org/
220) [LIGO] LIGO: Laser Interferometer Gravitational Wave Observatory http://www.ligo.caltech.edu/
221) [Macdonald02A] Alison Macdonald and Philip Lord, Digital Data Curation Task Force, *Report of the Task Force Strategy Discussion Day*, 26th November 2002, http://www.jisc.ac.uk/uploaded_documents/CurationTaskForceFinal1.pdf
222) [ManCSAR] CSAR: Manchester Computer Services for Academic Research (CSAR), http://www.csar.cfs.ac.uk/
223) [Mann03A] V. Mann and M. Parashar, *Discover and Computational Steering*, Chapter 31 of [Berman03A]
224) [Marinescu01A] Mini-workshop on Internet Process Coordination and Ubiquitous Computing, December 13-15, 2001, University of Central Florida, http://bond.cs.ucf.edu/workshop/. Published in C. Marinescu and Craig Lee (Eds). Process Coordination and Ubiquitous Computing, CRC Press 2002.
225) [Marinescu02A] Dan C. Marinescu, *Internet-Based Workflow Management: Toward a Semantic Web*, Wiley, April 2002, ISBN: 0-471-43962-2
226) [Marinescu02B] D. C. Marinescu, *A Grid Workflow Management Architecture*, Global Grid Forum White Paper, http://www.cs.ucf.edu/%7Edcm/GWfA.pdf
227) [Mathematica](↑) Mathematica http://www.wolfram.com/products/mathematica/index.html
228) [MATLAB](↑) MATLAB http://www.mathworks.com/
229) [Matthews00A] Matthews, W., Cottrell, L., *The PingER project: Active Internet Performance Monitoring for the HENP Community*, IEEE Communications Magazine, May 2000

230) [Matthews01A] Brian Matthews and Shoaib Sufi, *CLRC Scientific Metadata Format*, February 2001 technical report DL TR 02001, http://www-dienst.rl.ac.uk/library/2002/tr/dltr-2002001.pdf
231) [Mehrotra02A] P. Mehrotra, *Issues in Workflow Specifications & Management*, http://www.computingportals.org/meetings/ggf6/ppt/wf-issues-ggf6.ppt
232) [MIAKT](↑) MIAKT: Medical Imaging and Advanced Knowledge Technologies http://www.aktors.org/miakt/
233) [MichiganKX509](↑) Michigan University's KX509 Kerberized PKI Project http://www.citi.umich.edu/projects/kerb_pki/
234) [Mock02A] Steve Mock, Kurt Mueller, Marlon Pierce, Choonhan Youn, Geoffrey Fox, and Mary Thomas *A Batch Script Generator Web Service for Computational Portals*, Proceedings of IC-02 June 2002. http://grids.ucs.indiana.edu:9000/slide/ptliu/research/gateway/Papers/BSGCIC.pdf
235) [MOM](↑) Message Oriented Middleware http://www.middleware.org/mom/basicmom.html
236) [Mono.NET](↑) Mono Project bringing .NET to Linux http://www.go-mono.com/.
237) [Moore01A] Patrick C. Moore, Wilbur R. Johnson, Richard J. Detry, *Adapting Globus and Kerberos for a Secure ASCI Grid*, Sandia National Laboratory integration of Globus and Kerberos in Proceedings of SC01 November 2001, http://www.sc2001.org/papers/pap.pap192.pdf
238) [Moore03A] Reagan Moore and Chaitan Baru, *Virtualization Services for Data Grids*, Chapter 16 of [Berman03A]
239) [Morpheus](↑) Morpheus peer-to-peer System http://www.morpheus-os.com/ and http://www.musiccity.com/
240) [MpCCI](↑) MpCCI code coupling interface for multidisciplinary applications http://www.mpcci.org/
241) [MPICH-G2](↑) MPICH-G2 grid-enabled implementation of the MPI v1.1 standard based on the MPICH library http://www.nsf-middleware.org/NMIR3/components/mpichg2.asp
242) [MQSeries](↑) MQSeries in IBM WebSphere http://www-3.ibm.com/software/integration/websphere/services/
243) [Mueller02A] PACX-MPI described in M. Mueller , E. Gabriel and M. Resch, *A Software Development Environment for Grid Computing*, Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-15, pages 1543-1552, 2002.
244) [myGrid-A](↑) myGrid e-Science Bioinformatics Project http://myGrid.man.ac.uk/.
245) [myGrid-B](↑) myGrid Workflow http://mygrid.man.ac.uk/myGrid/web/components/Workflow/
246) [myGrid-C](↑) myGrid Notification service http://mygrid.man.ac.uk/myGrid/web/components/NotificationService/
247) [myGrid-D](↑) The myGrid Provenance Service http://mygrid.man.ac.uk/myGrid/web/components/ProvenanceData/
248) [MyProxy](↑) MyProxy credential repository for the Grid http://www.nsf-middleware.org/NMIR3/components/myproxy.asp
249) [Nacar03A] Mehmet Nacar, Marlon Pierce and Geoffrey Fox, *Designing a Grid Computing Environment Shell Engine* in Proceedings of the 2003 International Conference on Internet Computing, Las Vegas June 2003, http://grids.ucs.indiana.edu/ptliupages/publications/GCEShell_camera.pdf
250) [Nakada03A] Hidemoto Nakada, Yoshio Tanaka, Satoshi Matsuoka, Staoshi Sekiguchi, Tokyo and Tsukuba, *Ninf-G: a GridRPC system on the Globus Toolkit*, Chapter 25 of [Berman03A]
251) [NaradaBrokering] NaradaBrokering from Indiana University http://www.naradabrokering.org
252) [Natrajan02A] Anand Natrajan, Anh Nguyen-Tuong, Marty A. Humphrey and Andrew S. Grimshaw, *The Legion Grid Portal*, Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-15, 1365-1394, 2002.
253) [NCSA] NCSA: National Center for Supercomputing Applications http://www.ncsa.uiuc.edu/
254) [NCSAGAMS] NCSA Grid Application Management Service http://www.extreme.indiana.edu/alliance/miledeliv/
255) [NeesGrid-A](↑) NeesGrid Earthquake Engineering Grid http://www.neesgrid.org
256) [NeesGrid-B](↑) NeesGrid Metadata Service http://www.neesgrid.org/repository/
257) [NERCDataGrid](↑) NDG: NERC DataGrid http://ndg.nerc.ac.uk/
258) [NEReSC](↑) North-East Regional e-Science Centre http://www.neresc.ac.uk/index.html
259) [NetCDF] NetCDF: Network Common Data Form interface for array-oriented data access http://www.unidata.ucar.edu/packages/netcdf/
260) [Netwarfare](↑) Network-Centric Warfare http://www.c3i.osd.mil/NCW

261) [NetworkMonitor](↑) Collection of Network Monitoring Sites http://www.slac.stanford.edu/comp/net/wan-mon/netmon.html
262) [Nimrod](↑) Nimrod Grid Tool for Parametric Modeling, http://www.csse.monash.edu.au/~davida/nimrod.html/
263) [Ninf](↑) Ninf network server project http://ninf.apgrid.org/
264) [NinjaSDS] Service Discovery Service from Berkeley developed as part of the Ninja project http://ninja.cs.berkeley.edu/
265) [NMI](↑) NSF Middleware Initiative http://www.nsf-middleware.org/
266) [Novotny03A] Jason Novotny, *Grid Portal Development Toolkit*, Chapter 27 of [Berman03A]. This system is known as GPDK
267) [NSF03A] Report of the National Science Foundation Blue-Ribbon Advisory Panel, *Revolutionizing Science and Engineering Through Cyberinfrastructure*, http://www.cise.nsf.gov/evnt/reports/toc.htm
268) [NVO](↑) US National Virtual Observatory http://www.us-vo.org/
269) [NWS](↑) Network Weather Service NWS http://www.nsf-middleware.org/documentation/NMI-R3/0/NWS/index.htm
270) [OAI](↑) Open Archives Initiative http://www.openarchives.org
271) [OASIS] OASIS: Organization for the Advancement of Structured Information Standards http://www.oasis-open.org/home/index.php
272) [OASISWSRM] WSRM: OASIS Web services Reliable Messaging http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrm
273) [OGCE] OGCE: Open Grid Computing Environment Consortium http://www.ogce.org
274) [OGCWS](↑) OpenGIS Consortium Web Services Initiative http://ip.opengis.org/ows2/
275) [OGSA-DAI](↑) OGSA-DAI Grid and Web database interface http://www.ogsa-dai.org/
276) [OGSA-DAIS](↑) OGSA-DAIS Working Group of Global Grid Forum http://www.gridforum.org/6_DATA/dais.htm
277) [OGSA-DAIT] OGSA-DAIT proposed extension to OGSA-DAI http://www.ogsa-dai.org/
278) [OGSA](↑) Open Grid Services Architecture (OGSA) http://www.gridforum.org/ogsi-wg/drafts/ogsa_draft2.9_2002-06-22.pdf
279) [OGSI.netUVA](↑) OGSI.net Project from the University of Virginia, http://www.cs.virginia.edu/~gsw2c/ogsi.net.html
280) [OGSI](↑) OGSI Open Grid Service Infrastructure Working Group of Global Grid Forum http://www.gridforum.org/ogsi-wg/
281) [OPeNDAP](↑) OPeNDAP: Open-source Project for a Network Data Access Protocol http://opendap.org/
282) [OPenDAPg] OPeNDAPg: Grid enabled OPeNDAP http://dods.hao.ucar.edu/#grid
283) [OpenJavaMOP](↑) The OpenJava MOP (Metaobject Protocol) http://www.csg.is.titech.ac.jp/openjava/
284) [Oram01A] *Peer-To-Peer: Harnessing the Benefits of a Disruptive Technology*, edited by Andy Oram, O'Reilly Press March 2001
285) [OWL](↑) OWL Web Ontology Language: *Web Ontology Language (OWL) Guide Version 1.0*, M. K. Smith, D. McGuinness, R. Volz, and C. Welty. W3C Working Draft 4 November 2002. Available from http://www.w3.org/TR/2002/WD-owl-guide-20021104/.
286) [Pacman-A](↑) Pacman packaging manager http://physics.bu.edu/~youssef/pacman/
287) [Pacman-B](↑) VDT usage of Pacman http://www.lsc-group.phys.uwm.edu/vdt/why_pacman.html
288) [Parabon](↑) Parabon Java Desktop Grid http://www.parabon.com
289) [Pastry](↑) Pastry substrate for peer-to-peer applications from Microsoft, http://research.microsoft.com/~antr/Pastry/
290) [Pattnaik03A] Pratap Pattnaik, Kattamuri Ekanadham and Joefon Jann, *Autonomic Computing and GRID*, Chapter 13 of [Berman03A]
291) [Paxson96A] Paxson, V., *Towards a Framework for Defining Internet Performance Metrics*, Proceedings of INET'96, Montreal, Canada, June 1996
292) [PBS-A](↑) PBS: Portable Batch Scheduler http://www.openpbs.org/
293) [PBS-B](↑) PBS Workload Management http://www.pbspro.com/
294) [Permis](↑) Permis authorization system from Salford University UK, http://sec.isi.salford.ac.uk/permis/

295) [Pierce02A] Marlon Pierce, Choonhan Youn, Ozgur Balsoy, Geoffrey Fox, Steve Mock, and Kurt Mueller, *Interoperable Web Services for Computational Web Portals*, Proceedings of Supercomputing 2002. Baltimore (2002). http://grids.ucs.indiana.edu/ptliupages/publications/portalwsSC02.doc
296) [Pierce03A] Marlon Pierce and Geoffrey Fox, Federated Grids and their Security, http://grids.ucs.indiana.edu/ptliupages/publications/FedGrid_Short.pdf
297) [Placeware](↑) PlaceWare Web Conferencing http://www.placeware.com/
298) [PlaleWS](↑) Beth Plale, Indiana University, Relational Grid Resources, http://www.cs.indiana.edu/~plale/projects/RGR/
299) [Platform](↑) Platform Computing LSF for Enterprise Grids http://www.platform.com/products/wm/LSF/index.asp
300) [Polycom](↑) Polycom commercial video conferencing http://www.polycom.com
301) [PPDG](↑) PPDG Particle Physics Data Grid http://www.ppdg.net/
302) [PRISM](↑) PRISM: Programme for Integrated Earth System Model http://prism.enes.org/main.html
303) [PROMENVIR](↑) PROMENVIR wide area meta-computing demonstrations http://www.it-innovation.soton.ac.uk/services/eng_design/design_cases.shtml#promenvir
304) [PST](↑) PST: Practical Supercomputing Toolkit http://pstoolkit.org/
305) [Punch](↑) Punch Platform for Internet Computing from Purdue University http://punch.purdue.edu/HubInfo/about.html
306) [Pythia](↑) Pythia Problem Solving environment http://www.cs.purdue.edu/research/cse/pythia/index.html
307) [RDF-A] RDF: O. Lassila and R. R. Swick, eds. , Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.
308) [RDF-B] RDF Schema: D. Brinkley and R.V. Guha, eds., RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft 23 January 2003. http://www.w3.org/TR/rdf-schema/.
309) [RealityGrid](↑) RealityGrid e-Science Project http://www.realitygrid.org/
310) [Reid00A] Reid, A., Flatt, M., Stoller, L., Lepreau, J., Eide, E., *Knit: Component Composition for Systems Software*, OSDI 2000, pp347-360, Oct 2000.
311) [RIPENCC] RIPE NCC Test Traffic Measurements, http://www.ripe.net/ripencc/mem-services/ttm/index.html
312) [Roman00A] Roman, M., Mickunas, D., Kon, F., and Campbell, R.H., *LegORB*, Proc. IFIP/ACM Middleware'2000 Workshop on Reflective Middleware, IBM Palisades Executive Conference Center, NY, April 2000.
313) [Romberg02A] UNICORE Workflow in M. Romberg, *The UNICORE Grid. Infrastructure*, Scientific Programming, Special Issue on Grid Computing, Vol 10, No. 2, pp. 149-157, 2002.
314) [RPM](↑) RPM Package Manager http://www.rpm.org/
315) [SCECGrid](↑) Southern California Earthquake Center Grid http://www.scec.org/cme/
316) [SchmidtWS-A] Douglas C. Schmidt, Washington University St. Louis, Research on Fault-Tolerant and Dependable CORBA, http://www.cs.wustl.edu/~schmidt/corba-research-reliable.html
317) [SCIRun](↑) SCIRun http://software.sci.utah.edu/scirun.html
318) [SDSS] SDSS: Sloan Digital Sky Survey http://www.sdss.org/
319) [SDT] Southampton University Semantic Discovery Toolkit SDT. See appendix section A.2.6.1.5 by Luc Moreau of [GapAnalysis]
320) [SemanticGrid](↑) Semantic Grid http://www.semanticgrid.org
321) [SemanticWeb](↑) Semantic Web http://www.w3.org/2001/sw/
322) [SERVOGrid](↑) Solid Earth Research Virtual Observatory http://www.servogrid.org
323) [SETI](↑) SETI@Home Internet Computing http://setiathome.ssl.berkeley.edu/
324) [SGE](↑) Sun Grid Engine http://wwws.sun.com/software/gridware/
325) [Shibboleth](↑) Shibboleth Internet2 Project http://shibboleth.internet2.edu/
326) [Shum02A] Buckingham Shum, S., De Roure, D., Eisenstadt, M., Shadbolt, N. and Tate, A. (2002) *CoAKTinG: Collaborative Advanced Knowledge Technologies in the Grid*. Proceedings of the Second Workshop on Advanced Collaborative Environments, Eleventh IEEE Int. Symposium on High Performance Distributed Computing (HPDC-11), July 24-26, 2002, Edinburgh, Scotland
327) [Skyserver](↑) SkyServer: Sloan Digital Sky Survey data http://research.microsoft.com/~gray/SDSS/default.htm

328) [SlashGrid](↑) SlashGrid, a framework for Grid aware filesystems from GridPP, http://www.gridpp.ac.uk/authz/slashgrid/
329) [Slide](↑) Apache Slide Content Management System supporting WebDAV http://jakarta.apache.org/slide/
330) [Slominski02A] A. Slominski, Y. Simmhan, A. L. Rossi, M. Farrellee, and D. Gannon, XEvents/XMessages: *Application Events and Messaging Framework for the Grid*, Indiana University Computer Science Department Technical Report, June 2002, http://www.extreme.indiana.edu/xgws/papers/xevents_xmessages_tr.pdf.
331) [SLP](↑) IEEE Service Location Protocol (SLP) http://www.ietf.org/html.charters/svrloc-charter.html
332) [SmartFrog](↑) Hewlett Packard SmartFrog Configuration Framework http://www-uk.hpl.hp.com/smartfrog/
333) [SOAP](↑) SOAP: Simple Object Access Protocol http://www.w3.org/TR/SOAP/
334) [Spitfire](↑) Spitfire Relational Database middleware from European DataGrid Work Package 2, http://spitfire.web.cern.ch/Spitfire/
335) [SRBMCAT](↑) Storage Resource Broker (SRB) and Metadata Catalog (MCAT) from SDSC http://www.npaci.edu/DICE/SRB/index.html
336) [SRM](↑) SRM: Storage Resource Management Working Group http://sdm.lbl.gov/srm-wg/
337) [Surridge02A] Mike Surridge A Rough Guide to Grid Security version 1.1a, 11 September 2002, http://umbriel.dcs.gla.ac.uk/Nesc/general/technical_papers/RoughGuideToGridSecurityV1_1a.pdf
338) [SurveyorNetwork] The Surveyor Project Homepage, http://www.advanced.org/surveyor/
339) [SWFL](↑) SWFL: Service Workflow Language http://www.cs.cf.ac.uk/User/Yan.Huang/GridWF/SWFL.htm
340) [SWOF](↑) SWOF Scientific Workspaces of the Future http://www-unix.mcs.anl.gov/fl/research/SWOF/
341) [Syed02A] J Syed, M Ghanem, Y Guo, DISCOVERY PROCESSES: REPRESENTATION AND RE-USE, 2002 e-Science All Hands meeting, http://jameel.recoil.org/publications/allhands2002.pdf
342) [Szyperski98A] Szyperski, C., *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley, 1998.
343) [Terena-A](↑) TERENA - Trans-European Research and Education Networking Association http://www.terena.nl/
344) [Terena-B](↑) Report of the TERENA Technical Advisory Council, Zagreb, Monday 19 May 2003, http://www.terena.nl/tech/tac/zagreb/TAC-200305.pdf
345) [Thain03A] Douglas Thain, Todd Tannenbaum, and Miron Livny, *Condor and the Grid*, Chapter 11 of [Berman03A]
346) [Thomas03A] Mary Thomas and Jay Boisseau, *Building Grid Computing Portals: The NPACI Grid Portal Toolkit*, Chapter 28 of [Berman03A]
347) [TOOLSHED](↑) TOOLSHED problem solving environment http://www.it-innovation.soton.ac.uk/services/eng_design/design_cases.shtml#toolshed
348) [Triana-A](↑) Triana Project http://www.triana.co.uk/
349) [Triana-B](↑) Open Source Triana Software http://trianacode.org/
350) [Triana-C] Triana Workflow http://www.gridlab.org/WorkPackages/wp-3/D3.3.pdf
351) [Trillium](↑) Trillium Interlinked Grids http://www.hicb.org/TrilliumNewsletter/index.htm
352) [UDDI4J](↑) UDDI4J: Java UDDI support http://www-124.ibm.com/developerworks/oss/uddi4j/
353) [UDDI-A](↑) UDDI: Universal Description, Discovery and Integration technology from OASIS http://www.uddi.org/
354) [UDDI-B](↑) UDDI: T. Bellwood, et al. "UDDI Version 3.0 Published Specification, 19 July 2002" (2002). http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.
355) [UKeS-A](↑) UK e-Science Program http://www.escience-grid.org.uk/
356) [UKeS-B] CCLRC e-Science Centre http://www.e-science.clrc.ac.uk
357) [UKeS-C](↑) NeSC: National e-Science centre for UK program at Edinburgh, http://www.nesc.ac.uk/
358) [UKeSGridmon](↑) UK e-Science Grid Network Monitoring http://rtlin1.dl.ac.uk/gridmon/
359) [UKeSMarket](↑) Computational Markets UK e-Science project http://www.lesc.ic.ac.uk/markets/

360) [UKeSSDMIV](↑) Meeting on Scientific Data Mining, Integration and Visualization (SDMIV), 24 and 25 October 2002, e-Science Institute, Edinburgh, http://www.anc.ed.ac.uk/sdmiv/
361) [UKeSSTF] UK e-Science program security task force http://umbriel.dcs.gla.ac.uk/NeSC/general/teams/stf/
362) [UKeSViz] Meeting at NeSC Edinburgh January 23 2003 on Visualisation for e-Science http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=130
363) [Unicore-A](↑) UNICORE UNiform Interface to COmputing Resources http://www.unicore.de/
364) [Unicore-B](↑) Unicore Information Service as enhanced at Manchester University. See appendix section A.2.6.1.2 of [GapAnalysis]
365) [Unicore-C](↑) Unicore Forum http://www.unicore.org/forum.htm
366) [UnitedDevices](↑) United Devices http://www.ud.com/home.htm
367) [Uyar03A] XGSP Collaboration Web Service framework in Wenjun Wu, Ahmet Uyar, Hasan Bulut, Geoffrey Fox, *Integration of SIP VoIP and Messaging with the AccessGrid and H.323 Systems*, in Proceedings of 1st International Conference on Web Services Las Vegas June 2003, http://grids.ucs.indiana.edu/ptliupages/publications/sip-webservices-short02.pdf.
368) [VDT](↑) Virtual Data Toolkit from Griphyn [GriPhyn] http://www.lsc-group.phys.uwm.edu/vdt/contents.html
369) [VizServer] SGI Visualization Server http://www.sgi.com/software/vizserver/
370) [VRVS](↑) VRVS Virtual Rooms Video Conferencing System http://www.vrvs.org/
371) [W3C](↑) W3C: World Wide Web Consortium http://www.w3.org/
372) [Watson03A] Paul Watson, *Databases and the Grid*, Chapter 14 of [Berman03A]
373) [Watson03B] William A. Watson, Ying Chen, Jie Chen, Walt Akers, *Storage Manager and File Transfer Web Services,* Chapter 34 of [Berman03A]
374) [WebDAV](↑) WebDAV: *Web-based Distributed Authoring and Versioning*, http://www.webdav.org/
375) [WebEx](↑) WebEx commercial web conferencing http://www.webex.com
376) [WebSphere](↑) WebSphere: IBM Internet Application Server http://www-3.ibm.com/software/webservers/
377) [WfMC](↑) Workflow Management Coalition http://www.wfmc.org/
378) [WhiteRose](↑) White Rose Grid for the Universities of Leeds, Sheffield and York, http://www.wrgrid.org.uk/
379) [Williams03A] Roy Williams, *Grids and the Virtual Observatory*, Chapter 38 of [Berman03A]
380) [Wolski03A] Rich Wolski, Todd Bryan, James Plank, John Brevik, *Grid Resource Allocation and Control Using Computational Economies*, Chapter 32 of [Berman03A]
381) [WSAddressing](↑) Draft Web Service Addressing Standard from IBM and Microsoft http://msdn.microsoft.com/ws/2003/03/ws-addressing/
382) [WSCL](↑) WSCL Web Services Conversation Language http://www.w3.org/TR/wscl10/
383) [WSDL4J](↑) WSDL4J: IBM Web Services Description Language for Java Toolkit http://www-124.ibm.com/developerworks/projects/wsdl4j/
384) [WSDL](↑) WSDL: Web Services Description Language http://www.w3.org/TR/wsdl.html from [W3C]
385) [WSFL](↑) WSFL: Web Services Flow Language http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
386) [WSIF](↑) WSIF: Apache Web Services Invocation Framework http://ws.apache.org/wsif/
387) [WSIL](↑) WSIL: K. Ballinger, P. Brittenham, A. Malhotra, W. A. Nagy, and S. Pharies, "Specification: Web Service Inspection Language (WS-Inspection) 1.0." (2001). http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html.
388) [WSReliableMessage](↑) Draft Web Service Reliable Messaging Standard from BEA IBM Microsoft and TIBCO http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-reliablemessaging.asp or http://www-106.ibm.com/developerworks/library/ws-rm/
389) [WSRP](↑) OASIS Web Services for Remote Portlets (WSRP) http://www.oasis-open.org/committees/
390) [XPDL](↑) XPDL XML Processing Description Language http://www.ebpml.org/xpdl.htm