Decision Engineering Report Series
Edited by Rajkumar Roy and Yuchun Xu

GRID COMPUTING FOR ENGINEERING DESIGN OPTIMISATION: EVOLUTION AND FUTURE TRENDS

Gokop Goteng, Ashutosh Tiwari and Rajkumar Roy

March 2007

Cranfield University
Cranfield
Bedfordshire
MK43 OAL
United Kingdom

http://www.cranfield.ac.uk

ISBN

'*Decision Engineering*' is an emerging discipline that focuses on developing tools and techniques for informed operational and business decision-making within industry by utilising data and information available at the time (facts) and distributed organisational knowledge.

The '*Decision Engineering Report Series*' from Cranfield University publishes the research results from the ***Decision Engineering Centre*** in Enterprise Integration. The research centre aims to establish itself as the leader in applied decision engineering research. The client base of the centre includes: Airbus, BAE SYSTEMS, BOC Edwards, BT Exact, Corus, EDS (Electronic Data Systems), Ford Motor Company, GKN Aerospace, Ministry of Defence (UK MOD), Nissan Technology Centre Europe, Johnson Controls, PRICE Systems, Rolls-Royce, Society of Motor Manufacturers and Traders (SMMT) and XR Associates.

The intention of the report series is to disseminate the centre's findings faster and with greater detail than regular publications. The reports are produced on the core research interests within the centre:

- Cost Engineering
- Product Engineering
- Applied soft computing

Edited by:

Professor Rajkumar Roy
r.roy@cranfield.ac.uk

David Baxter
d.baxter@cranfield.ac.uk

Enterprise Integration
Cranfield University
Cranfield
Bedfordshire
MK43 OAL
United Kingdom

http://www.cranfield.ac.uk

Series librarian:

John Harrington
j.harrington@cranfield.ac.uk

Kings Norton Library
Cranfield University

Publisher: Cranfield University

Abstract

Grid Computing is fast gaining ground both within academia and the commercial sectors. It has shifted from its traditional scientific-based applications to service-oriented problem solving environments for commerce and business. Engineering design optimisation (EDO) is characteristically computationally and data intensive. EDO is also a multidisciplinary field which requires the collaboration of different domain experts to work on a design to yield improved versions. Grid Computing offers a suitable platform for design engineers to collaboratively work together and share knowledge and expertise in addition to the computational and data facility that can be combined to bear on complex designs. In this paper, the trend of Grid Computing evolution shows a clear emergence of application areas, starting from computational grid, data grid, visualisation grid and semantic grid to service-oriented problem solving environments (SO-PSE). This evolution is classified as first, second and third generation of Grid Computing for the purpose of understanding how researchers have tried to provide solutions to the problems and challenges in implementing Grid applications. The future of Grid Computing research areas such as autonomic computing, ubiquitous computing and economic Grid models as well as concurrent engineering design problem solving environments feature in the report. Autonomic computing enables grid services and resources to have self-management, self adjustable and adaptability to changing and dynamic situations using agent-based technology while ubiquitous computing allows computers to perceive the environment and act accordingly.

Keywords: Engineering design optimisation, Grid computing, Problem solving environments, Service-oriented architecture.

Table of Contents

Normenclature

Abbreviation	Meaning
CA	Certificate Authority
DAME	Distributed Aircraft Maintenance Environment
DECGrid	Decision Engineering Centre Grid
EC	Evolutionary Computing
EDO	Engineering Design Optimisation
EP	Evolutionary Programming
ES	Evolutionary Strategy
FIPER	Federated Intelligent Product Environment
GA	Genetic Algorithm
GEODISE	Grid-Enabled Optimisation Design Search for Engineers
GIS	Grid Information Service
GP	Genetic Programming
GRAM	Grid Resource Allocation Management
GRIA	Grid Resource for Industrial Application
GridFTP	Grid File Transfer Protocol
GSDL	Grid Services Description Language
GSH	Grid Service Handle
GSI	Grid Security Infrastructure
GSR	Grid Service Reference

GT	Globus Toolkit
I-WAY	Information Wide Area Year
OGF	Open Grid Forum
OGSA	Open Grid Service Architecture
PSE	Problem Solving Environment
QoD	Quality of Data
QoS	Quality of Services
SDE	Service Data Element
SDSC	San Diego Supercomputing Centre
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SORCER	Service-Oriented Computing Environment
SRB	Storage Resource Broker
UDDI	Universal Description, Discovery and Integration
VO	Virtual Organisation
WebMDS	Web Monitoring and Discovery Service
WSDL	Web Services Description Language
WSRF	Web Services Resource Framework

1. Introduction

The competitive nature of the global economy is responsible for adoption of various technologies by companies to manufacture superior products so as to gain and retain a better share of the market over their competitors. At the fore front of this is the optimisation of engineering designs. Engineering design optimisation (EDO) is the process whereby engineering modeling and analysis tools are exploited and used to yield technically and economically improved designs. EDO usually involves the coupling together of computer aided design (CAD) tools, analysis codes for computational fluid dynamics (CFD), finite element analysis (FEA), and genetic algorithms (GAs) as search tools (Cox et al., 2002). Traditionally, EDO is performed using parallel distributed supercomputing (PDSC) technology as a platform due to its computation, data intensive and multidisciplinary nature. However, the dynamic addition and removal of services in EDO processes coupled with other requirements that deal with real time heterogeneity, coordination, and security make PDSC time consuming, costly and in some cases not possible. This is where Grid Computing comes in. Grid Computing (GC) is a distributed large-scale computing infrastructure which offers secured, pervasive, dependable, transparent, inexpensive, and coordinated resource sharing (Foster and Kesselman, 1999). Resource here means hardware, software, data, information, knowledge, computational and optimisation codes, and visualisation instruments among other things. Grid computing toolkits and middleware provide suites of services that enhance the job of design engineers. For example, the Geodise (Grid Enabled Optimisation Design Search for Engineering) toolkit and Globus toolkit (Eres et al., 2003) provide problem solving environments for Grid-enabled parametric generation, meshing, CFD analysis, design optimisation and search, and visualisation for performing EDO. With these tools made handy to engineers over a Grid, there may be a reduction in design cycle time and reduction in time to market. This report intends to present the current status of grid research by tracing its origin and evolution. The report puts together the state-of-the-art evolution and future trend in grid computing research. This evolution is classified as first, second, third and future generations of grid computing. This classification is based on different emphases given to different application areas as researchers gained more insight into grid capabilities and discovered more potentials for its global usage. This document is divided into three main parts. The first part describes grid computing and its architecture as well as engineering design optimisation (EDO) and how grid can be used to make EDO more efficient. The second part describes generations of grid computing namely first, second and third generations and they relate to computational, data, knowledge, collaboration and visualisation needs of EDO as the grid evolves. The last part describes future grid applications and challenges. Such future applications include autonomic computing and ubiquitous computing. Another great challenge in the grid architecture that needs to be addressed before it is widely accepted by industries is security.

1.1. Grid computing

The state of play in the metamorphosis of Grid computing is reminiscent of programming languages in the early 60s. From a purely mathematical and scientific tool for solving complex scientific problems by mathematicians and scientists using machine coded and assembly languages to a more friendly, human-like procedural and later object-oriented (OO) languages for solving mathematical, scientific, commercial, political, and almost every day human activities that will bring automation and efficiency. Just like programming languages which started their evolution with the first generation languages to the present fifth generation languages, the Grid is now in its third generation of evolution and development. This trend will undoubtedly continue in the coming years. In its evolution, the Grid has developed from computational and data intensive platform for solving large-scale science problems to a service-oriented problem solving environment (SO-PSE) for solving multidisciplinary problems. This means that more Grid application areas are being discovered as the Grid matures with time. Engineering design optimisation which is computationally and data intensive and requires the collaboration and sharing of knowledge by different design engineers is suitable for Grid application. Additionally, engineers need to have easy access to design tools. In Grid, these tools can be published as services by service providers and consumed by different users. One example of a grid service is visualisation service which enables the design experts to have a real time multidimensional visual display of the design and make instant decisions.

The Grid has a layered architecture (Figure 1). The lowest layer (Fabric Layer) implements low level services and the upper most Layer (Application Layer) implements high level services. Low level services are services that concern the hardware and complex systems implementation while high level services concern the user interface and simple application implementation services. These characteristics make Grid computing attractive to application areas that require different levels of services to accomplish a task. For example, engineering design optimisation requires interaction with different operating systems and network protocols (low level services) to connect with other design engineers located elsewhere and also requires the use of CAD systems and sharing of optimisation codes (high level services) for design optimisation activities. Figure 1 shows the architecture of Grid computing. The collective and resource layers have services for application programmer's interfaces (APIs) and software development kits (SDKs). The connectivity layer has protocols to connect to other Grid platforms and networks using Transmission Control Protocol/Internet Protocol (TCP/IP) and other protocols.

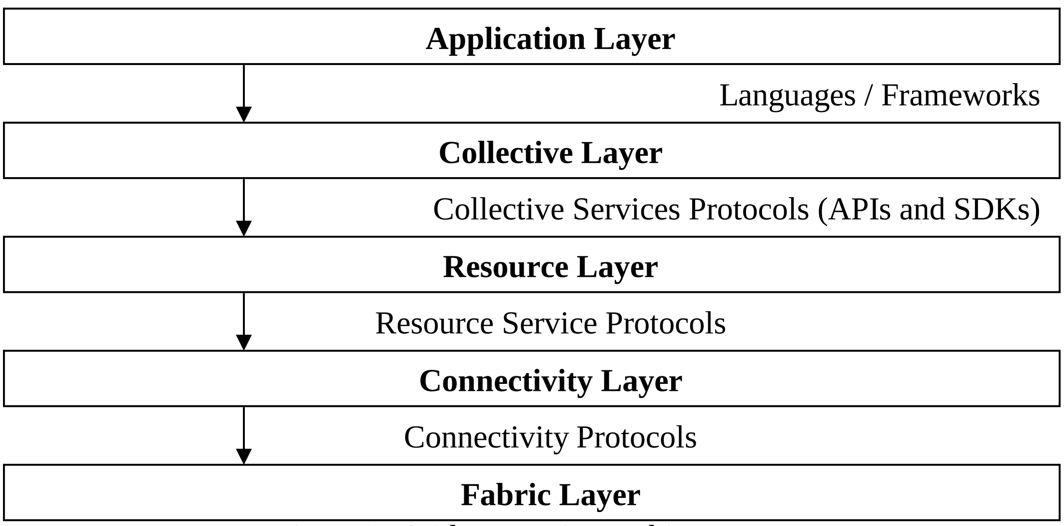

Figure 1: Grid computing architecture

1.2. Engineering design optimisation

The conventional model for engineering design optimisation usually starts by accepting input variables, performing computation on the objective function and the constraints, producing results and returning the results to the optimisation engine. The optimisation engine then filters the best or optimum result and decides whether to terminate the optimisation process if the result is good enough or to send the result as an input for another round of the process if the result is not good enough.

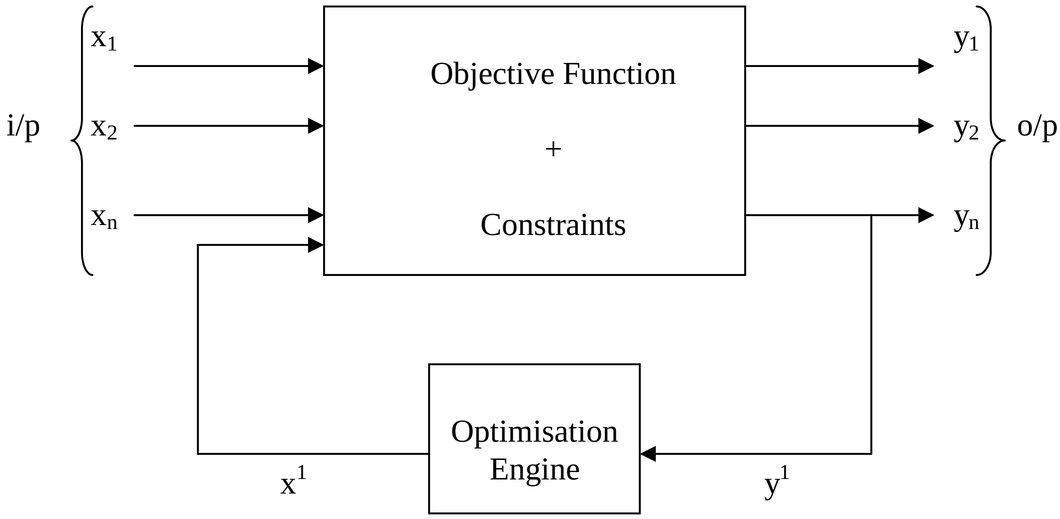

Figure 2 Optimisation model

This iterative process can go through many numbers of loops. In fact the cycle of iteration could be in millions when it involves scientific and engineering jobs that are computationally and data intensive and could take days or weeks to complete. In this process, data is generated in terabytes and Petra bytes in a variety of formats (Wason et al., 2004) which are heterogeneous in nature and may not be compatible with them. With the computational and data Grids, these computations and data generated can be performed and stored at different sites or nodes of the Grid. In this way, better optimisation results that would yield economically and technically optimum designs at faster rates are expected. Figure 2 is an example of an optimisation model with inputs (i/p), objective function, constraints, outputs (o/p) and optimisation engine. Toolkit such as the Geodise uses the capabilities of Grid computing and Web services as an opportunity for the design engineers to access an extended range of computing resources and manage the sizeable data created by such distributed applications (Wason et al., 2004) more economically and efficiently. However, even though the Grid could accomplish computational and data efficiency, the heterogeneous nature of the data formats generated and the software and hardware managing the computation and data at the different nodes present a bottleneck in terms of poor interoperability. This again calls for the inclusion of middleware service. Middleware such as the Globus (developed from Information Wide Area Year (I-WAY)) is the first to be used in early Grid implementation.

The pervasiveness of collaboration is a great booster to technological advancement and knowledge. Decision making is an integral part of engineering profession especially in engineering design optimisation. The Grid provides this pervasive infrastructure for collaborative engineering optimisation decision making with interactive sessions. Apart from the technical issues such as maintaining the integrity of the databases when any changes are made (Lu, 2003) and the knowledge shared among the engineers, concurrent engineering (CE) methodology is encouraged too. The Grid Computing community has developed some test-beds to demonstrate the ubiquitous nature of grid computing. Ubiquitous Supercomputing Testbed Organisation (GUSTO) has some of the largest supercomputers in the world for its research and operations. GUSTO test beds do this by embedding small, inexpensive and robust networked devices that are aware of the physical effects of their environments. For example the project called *Things that Think* at Massachusetts Institute of Technology works on GPS (Geographical Positioning System) and mobile phones that locate their environments as well as direct behaviors of the systems.

Figure 3 shows the application areas in Grid computing and the distribution of 72 research papers obtained in different areas. Most of the research papers dealt with more than one application areas, showing intersections in the Venn diagram (Figure 3). Figure 3 is obtained from Table 1. Table 1 shows the classification of research papers and the author(s) and years of publication in different Grid application areas. Ten key phrases which contain words like grid computing, engineering design optimisation, problem solving environment and service-oriented architecture were used to search for the papers. The databases searched from include ScienceDirect/Compendex, Google Scholar, Cranfield University databases such as Elsevier, IEEE (Institute of Electrical/Electronics Engineers), Computer Machines, AIAA (American Institute of Aeronautic and Astronauts), Inspects and books among others. The search was performed from 1980 to 2006 year of publications. The search concentrated on grid computing areas that relate to engineering design optimisation. It does not include general areas in grid computing.

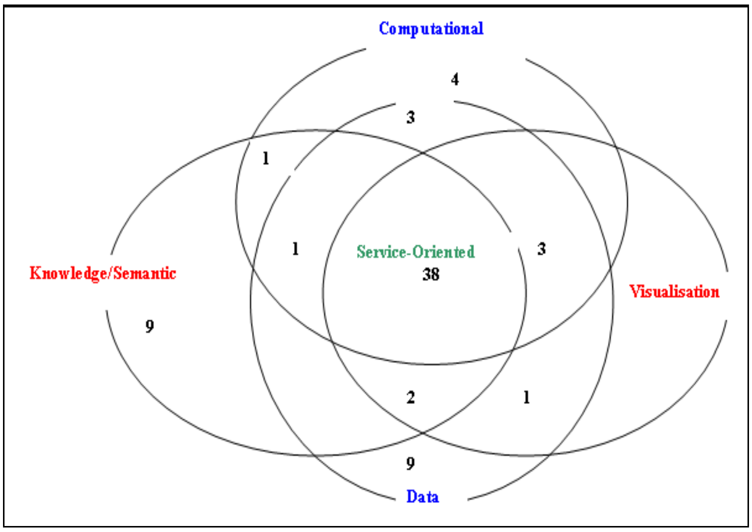

Figure 3: Grid application areas and number of research papers

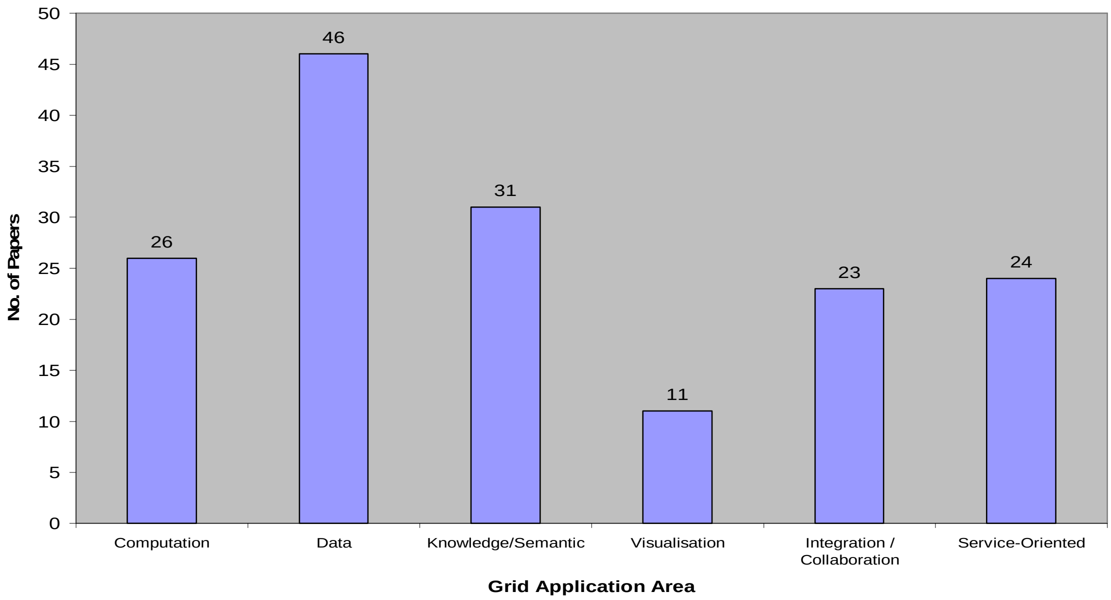

Figure 4: Grid application areas and number of research papers

Figure 4 shows Grid application areas and some selected research publications obtained during this study. It can be seen from the bar chart (Figure 4) that data application received more attention from the research papers than other application areas. This goes to show that data occupies an important position in the Grid evolution. This perhaps is because different organisations collaboratively use different databases with different schemas generating lots of metadata that need to be interpreted. Visualisation received the least attention. This shows that Grid visualisation at the time of this research is still at its infancy. Computational Grid occupies the second position in the chart (Figure 4) trailing behind Data Grid. Computation services are more developed than any service in the Grid evolution. This is because computation formed the first concept of Grid computing.

2. Generations of grid computing for engineering design optimisation

There are first, second and third generations of Grid computing. The first generation deals with computational and data intensive applications. The second generation deals with protocols and middleware to overcome interoperability problems among Grid users. The third generation is concerned with problem solving environments (PSE) and service-oriented architectures (SOA). Figure 5 shows the 3 generations of Grid computing and future research areas.

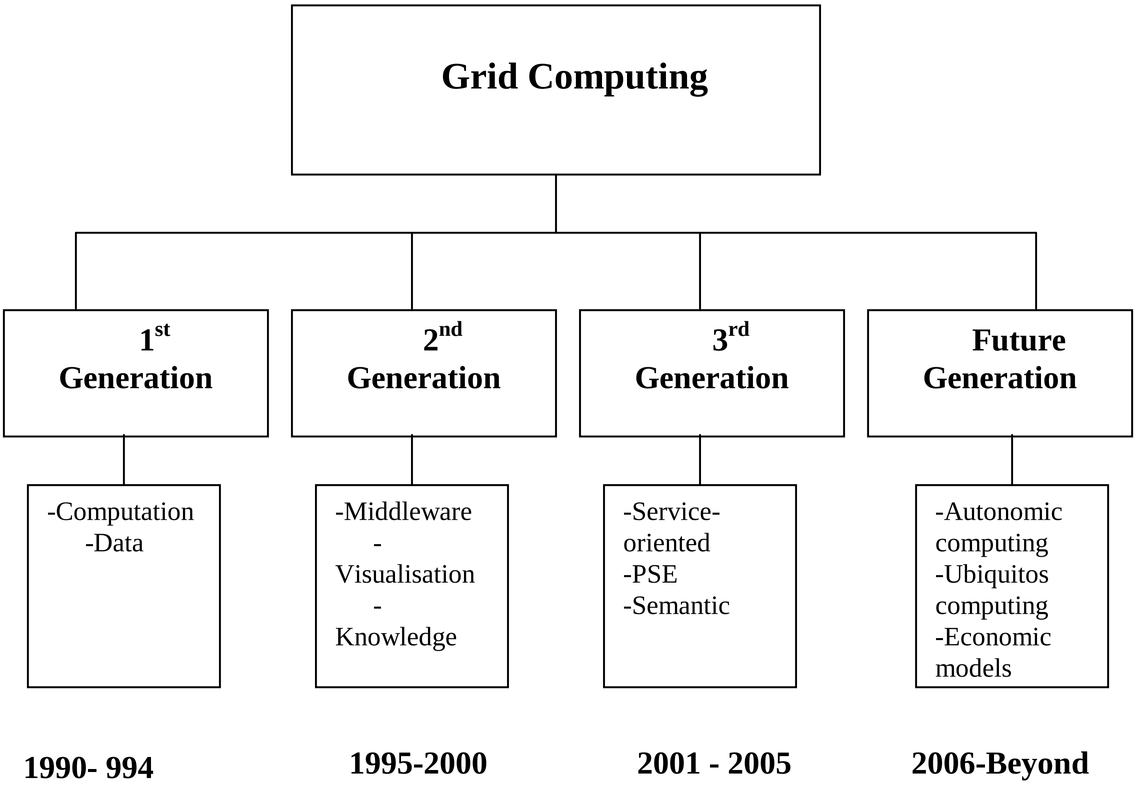

Figure 5: Grid computing generations and application areas

2.1. First generation of grid computing

The first generation of Grid computing is basically linking of different supercomputing resources to produce computational synergy among users at different sites.

2.1.1. Computational grid

In 1993, the National Science Foundation (NSF), National Computational Science Alliance (NCSA) in Illinois and the San Diego Supercomputer Centre (SDSC) in California brought together a number of supercomputing stations together to share computing power. This project proved to have a great synergistic effect. The result of this was the use of redundant computing cycle-time, faster computation, better computational results, cheaper maintenance cost, and effective throughput. This formed the first generation of Grid computing application which is popularly referred to as metacomputing (Berman et al, 2003). The whole idea of metacomputing is to harness the computing power that was ‘littered’ around at different geographical locations within the United States supercomputing stations. This is because none of them was able to single-handedly solve the scientific computations that involve computational fluid dynamics (CFD) and astrophysics computations. It was purely for scientific purpose among researchers and the academia. This effort to an appreciable extent solved the computational and data intensive application domain problems. The research papers that concentrated mainly on computational and data intensive aspects of the Grid as a motivation for Grid computing fall under the first generation research activities (Table

1). However, other problems cropped in. There was the problem of interoperability among the linked stations because of the heterogeneity of the resources (hardware and software) each is using. There was also lack of real collaboration and visualisation of results coupled with a lack of knowledge/semantic driven Grid that will ensure reuse. This takes the research to the next level-the second generation of Grid computing research activities. Table 1 shows the list of research papers based on application areas. The big ticks showed against application areas means that the paper placed much emphasis on that particular area and the small ticks mean less emphasis on the application area. An empty cell means that the research paper did not mention that particular application area. The application areas are arranged from computational to service-orientation and the emphasis progresses in that order as can be seen in the big ticks mean that the paper placed much emphasis on the corresponding application area. Figure 1 is obtained from Table 1.

S/N	Author(s)	Computationally Intensive	Data Intensive	Knowledge / Semantic	Visualisation	Integration / Collaboration	Service-Oriented
1	Goux et al (2000)	✔					
2	Phan et al (2002)	✔				✓	
3	Wolski et al (2001)	✔					
4	Shan et al (2003)	✔					
5	Getov et al (2001)	✔					
6	Buyya et al (2000)	✔				✓	
7	Venugopal, et al (2004)	✓	✔				
8	Wan et al (2003)		✔				
9	Foster et al (2002)	✓	✔				
10	Fox (2003)		✔	✓			
11	Rajasekar (2003)		✔			✓	
12	Lamehamedi et al (2002)		✔				
13	Cannataro et al (2002)	✓	✔	✓			
14	Bell et al (2003)		✔				
15	Kurata et al (2003)		✔				
16	Antonioletti et al (2003)		✔			✓	
17	Stockinger, (2002)		✔				
18	Deelman et al (2004)		✔				
19	Deelman et al (2002)	✓	✔				

20	Kaczmarski et al (2004)		✓				
21	Deelman et al (2001)		✓				
22	Rajasekar et al (2002)		✓				
23	Vazhkudai and Schopf (2002)		✓				
24	Kosar and Livny (2004)	✓	✓				
25	Antonioletti et al (2005)		✓				
26	Barbera et al (2003)		✓			✓	
27	Aloisio et al (2004)		✓				
28	Gil et al (2004)			✓		✓	
29	Hau et al (2003)	✓		✓			
30	Schwidder et al (2005)			✓		✓	
31	Fox (2003)			✓		✓	
32	Chen et al (2004)	✓		✓			
33	Zhuge (2005)			✓			
34	Canntaro and Talia (2004)	✓		✓			
35	Goble and Roure (2002)			✓			
36	Roure et al (2005)	✓		✓		✓	
37	Zhuge and Liu (2003)			✓		✓	
38	Zhuge (2002)			✓			
39	Pouchard et al (2003)			✓		✓	
40	Zhuge (2004)			✓			
41	Ghanem et al (2002)	✓		✓		✓	
42	Ghanem et al (2002)			✓			
43	Cannataro et al (2001)			✓			
44	Li and Lu (2004)	✓		✓		✓	
45	Schikuta and Weishaupl (2004)			✓			
46	Cannataro and Talia (2003a)			✓			

47	Zhuge and Shi (2004)			✔			
48	Cannataro and Talia (2003b)	✓		✔		✓	
49	Blythe et al (2003)			✔		✓	
50	Boose (1989)			✔			
51	Foster et al.(1999)	✓			✔	✓	
52	Norton and Rockwood (2003)	✓			✔		
53	Bethel and Shalf (2003)	✓			✔		
54	Kranzlmuller et al (2002)			✓	✔		
55	Brodlie et al (2004)				✔	✓	
56	Nielson et al (1994)			✓	✔	✓	
57	Shalf et al (2003)				✔	✓	
58	Brodlie et al (2004)	✓			✔		
59	Pearlman et al (2002)					✓	✔
60	Venugopal et al (2004)	✓					✔
61	Agarwal and Parashar (2003)						✔
62	Foster et al (2002)			✓		✓	✔
63	Barmouta and Buyya (2003)	✓					✔
64	Kacsuk et al (2004)					✓	✔
65	Hwang and Aravamudham (2004)	✓					✔
66	Stevens et al (2003)			✓			✔
67	Talia (2002)	✓		✓			✔
68	Smiles et al (2003)			✓			✔
69	Welch et al (2003)						✔
70	Papazoglou et al (2003)						✔
71	Andreozzi et al (2003)						✔

Table 1: Classification of Research Papers Based on Application Areas

[1] The big tick in Table 1indicates that the paper places much emphasis on the application area(s) and the small tick indicates less emphasis

Application areas which benefited from metacomputing project apart from the scientific and research communities are engineering, oil well reservoir exploration, data mining, climatic modeling, data modeling, and design optimisation. Engineering design optimisation (EDO) is naturally computational and data intensive. This is why numerical simulation is usually adopted for EDO. A major draw back to this process is the usual inadequate computational resource (Goodyer et al., 2005) for efficient design optimisation which might run into terabytes or even more that one single organisation might not be able to justify in a cost/benefit analysis. The conventional way companies perform engineering design optimisation is by using the local computing power they have at their disposal. The use of high performance supercomputers is very expensive and not cost effective for scientific research applications that involve computational and data intensive processes in the early phase (first generation of grid computing) of grid computing within academia. The introduction of computational Grid is a significant milestone that addressed this problem. For example, the UK e-Science programme for bioinformatics, particle physics and data access and integration (DAI) allows universities such as Cambridge, Imperial College London, Newcastle, Edinburgh and Oxford to collaborate and share computational resources. In this way, idle computing power is utilised efficiently and maintenance of different resources is shared. With Grids, the optimisation processes can be sub-divided into smaller tasks and launched separately to different nodes of the Grid at different Grid centre perhaps owned and maintained by the different (erstwhile supercomputing centre) organisations. The results of the different nodes can be pulled together to yet another node for final analysis and decision-making. In this way, the use of computational Grid, either across a single large enterprise or between many enterprises provides (Goodyer et al., 2005) a great deal of opportunities for cost-effective access and usage of large-scale computational resources with very good throughput and results that ultimately yield the desired and improved design. An additional advantage of using the Grid is in load balancing. The idle CPU cycle-time that could have been wasted under traditional parallel computing architecture is efficiently utilized by all Grid users at different peak times (Cao et al, 2003). Cao et al (2002) used agent-based technology to achieve dynamic load balancing in Grid applications that are computationally expensive through a performance-driven task scheduling algorithm. Grid scheduling middleware such as Condor also ensures high level of throughput for jobs launched at different nodes from geographically separated locations. DESY (Deutsches Elektronen Synchrotron), a German Electron Synchrotron and also a world leading research centre in particle accelerators achieved great throughput through Grid computing. The Monte Carlos simulations were handled by nodes of its collaborators located far away while the data and analysis aspects were handled by nodes locally located within the company that are part of the Global Grid (Gellrich et al., 2005).

2.1.2. Data grid

Data Grids have evolved to cater for the dual challenges of large datasets and multiple data repositories at distributed locations in data-intensive computing environments (Venugopal et al, 2004). Venugopal et al (2004) described how large communities of researchers around the world are engaged in the analysis, collation, and collections of data generated by scientific instruments and replicated on distributed resources. For example, applications for engineering design optimisation environments need to have Grid-enabled resource brokers that will allow design engineers have access to distributed data and computational resources. The resource brokers help to progressively discover, select, and publish required resources in an intelligent manner. Though the Grid enables the aggregation and sharing of resources among design experts, harnessing the power of grids from different nodes and sites of a Grid is still a challenging problem due to the complexity involved in the dynamic creation and heterogeneous composition of resources. The San Diego Supercomputing Centre (SDSC) uses Storage Resource Broker (SRB) as a data Grid middleware to integrate various types of data ranging from digital libraries to persistent archives stored in various databases and formats (Wan et al, 2003). SRB provides a storage repository abstraction for transparent and seamless access to multiple types of storage resources distributed globally. This is one of the innovative ways of overcoming the interoperability problems of the heterogeneous and dynamic Grid resources. Brokering efficiency can be enhanced by the information-rich nature of Grid services and resources.

Data intensive applications have received significant attention from Grid researchers. This is evident as can be seen in Figure 5 and Table 1. This is because the computational Grid depends on the availability of reliable data and data sources to do the computation. The creation of more data and metadata is important in engineering design optimisation.

In any computational setting, there is always a relationship among application programs, computation and data. The execution of programs performs computations using certain parameters or data and produces results in form of another data. This is to say that much scientific data is not obtained from measurements but rather derived from other data by the application of computational procedures (Foster et al., 2002a). This is typical of engineering design optimisation processes. In this case, there is the need to track the sequence of data generated and store the data derived at each step with unique identification. This will enable easy reconstruction/destruction of a particular portion of data generated during design optimisation. Foster et al. (2002b) developed a virtual data catalogue tool called Chimera which is based on the relational virtual data schema. Relational virtual data schema is a means of getting data that is generated from remote computational sources and used by different distributed databases to provide an expressive representation of computational procedures used to derive data as well as invocations of the procedures and the datasets produced by those invocations. This is a clever way of sharing and managing the knowledge obtained in metadata among design engineers working collaboratively. Figure 6 shows the relationships of programs, computation, data and how metadata can be created.

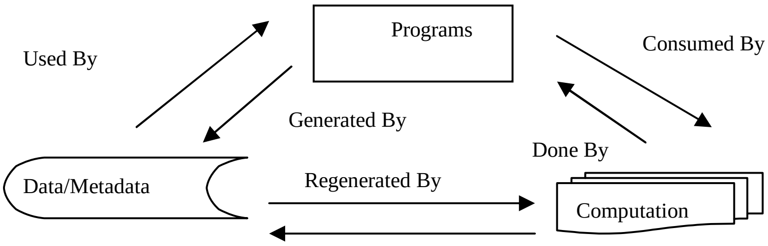

Figure 6: Data and metadata creation through program execution and computation.

2.1.3. Metadata

Metadata is an important part of the Grid regardless of whether one is considering Grid from its original inception of metacomputing (that is Grids linking distributed supercomputers) or the present semantic and service oriented Grid model (Fox, 2003). The term service metadata is becoming a critical feature of the evolving Grid architecture and data Grid. At the initial concept of Grid services, only computers and software were considered as services, but presently databases, sensors, networks, and user information systems are part of the Grid services. One great challenge in engineering design optimisation and analysis is to identify and locate the data and its subset data objects that are of interest to the design exploration process (Deelman et al, 2004). It is even more complicated in a Grid environment where data is dynamically created by many applications in the range of terabytes and petabytes. One way to overcome this problem is to describe the characteristics of each data object with one or more attributes known as metadata and to use this metadata as the means of identifying the relevant data objects (Deelman et al, 2004). The tools and resources including metadata requirements of a complex engineering design optimisation demands the use of highly coordinated management protocol that can autonomously achieve efficient data access, data publication and data discovery. In the GriPhyN (Grid Physics Network) project (Deelman et al, 2002), it was demonstrated that scientists can use Grid-enabled environments to seamlessly access data as raw data or metadata. In this way, data handling and processing capabilities can be integrated transparently to deliver data products to end-users or applications. Requests by design engineers for optimisation tools and services can then comfortably be mapped into computation or data at multiple locations around the world.

The term “Data Grid” traditionally stands for networks of distributed storage resources such as archival systems, databases, and caches that are globally linked using a logical name space that creates persistent identifiers which provide uniform and seamless access to users (Rajasekar et al., 2003).

2.1.4. Examples of EDO applications in first generation grid

Grid computing did not make much impact in EDO in the first generation. This is partly because the concept originated from academia and the first generation grid was used for science and engineering research purposes. The I-WAY project was purely to improve computational power by linking various supercomputing centres together. However, the early version of Globus toolkit (GT1) has features such as computation, data, resource management and information which are essential in an EDO environment.

2.2. Second generation of grid computing

The second generation of Grid computing concerns itself with communication protocols and middleware issues to solve interoperability related problems among collaborating supercomputing stations.

2.2.1. Middleware

The immediate problem faced by metacomputing (first generation Grid) era was interoperability. Having successfully linked the different supercomputing centre to harness computing power, the heterogeneous components (software and hardware) of the different centre hindered smooth communication. This problem formed the main discussion topic by the Global Grid Forum (GGF) which is now merged with the Enterprise Grid Alliance (EGA) to become the Open Grid Forum (OGF). The middleware concept to ensure interoperability among the supercomputer centre is the first initiative towards the second generation of Grid Computing. At the GGF meeting in 1993, Professor Ian Foster and Professor Carl Kesselman were charged with the responsibility to head a team to design and develop a robust Grid middleware. In 1995, the first tested Grid middleware, called the Information Wide Area Year (I-WAY) was demonstrated by 17 supercomputer centre and other computational research laboratories at the supercomputer '95 (SC95) in San Diego. I-WAY later evolved into the most popular Grid middleware (Berlich et al., 2005) called the Globus Toolkit (GT). The middleware aims at solving the interoperability problem witnessed in the first generation of Grid testbeds. Table 2 is a list of papers that focused attention on middleware research to bring together compute power, data, visualisation, and semantic/knowledge resources with regard to engineering design optimisation.

				Application Area(s)				
S/N	Author(s)	Service – Orientation Model / Architecture	Middleware	Compute	Data	Knowledge / Semantic	Visualisation	Engineering / Optimisation
1	Parashar et al (2004)	CORBA CoG	STORM / Discover /Pawn / Globus	✓	✓	✓	✓	✓
2	Eres et al (2003)	Geodise / Matlab	Globus	✓	✓	✓		✓
3	Pound et al (2003a)	Geodise / Matlab	Globus	✓	✓	✓		✓
4	Sobolewski and Kolonay (2006)	GISO / FIPER	Sun Microsystems Jini	✓	✓	✓	✓	✓
5	Tao et al (2004)	Geodise / OGSA	Globus	✓	✓	✓	✓	✓
6	Parmee et al (2005)	SEO	Triana	✓	✓	✓	✓	✓
7	Eres et al (2003)	Geodise / Matlab	Globus	✓	✓	✓		✓
8	Grauer et al (2004)	OpTiX	Globus	✓	✓	✓	✓	✓
9	Eres et al (2004)	Geodise / Matlab	Globus / Condor	✓	✓	✓		✓
10	Xu et al (2004)	Geodise / Matlab	Globus	✓				✓

11	Dai et al (2005)	OGSA	Globus					✓
12	Li et al (2006)	OGSA / OGSI	Globus				✓	
13	Ong et al (2005)	OGSA / DAME	Globus			✓		✓
14	Kao et al (2004)	FIPER	RMI/COM/JNI/DCOM	✓				✓
15	Teranishi et al (2004)	OGSA	Globus	✓				✓
16	Sobolewski (2004)	FIPER	Globus	✓				✓
17	Goel et al (2005)	FIPER	Globus	✓	✓			✓
18	Mahon et al (2003)	OGSA	Globus	✓	✓		✓	✓
19	Sobolewski et al (2003)	FIPER / SORCER	Globus		✓			✓
20	Shimosaka et al (2005)	OGSA	GridRPC	✓			✓	✓
21	Rohl et al (2000)	FIPER	Sun Microsystems Jini			✓	✓	✓
22	Soorianarayanan and Soblewski (2004)	SORCER	Globus			✓		✓
23	Cox et al (2001)	OGSA	Condor			✓		✓
24	Song et al (2004)	Geodise / OGSA	Globus			✓		✓
25	Xue et al (2004)	OGSA	JNI / OGSI.NET					✓
26	Pound et al (2003b)	Geodise / Matlab	Globus					✓
27	Song et al (2004)	Matlab / OGSA / OGSI	Globus					✓
28	Pound et al (2003c)	Geodise / Matlab	Globus					✓
29	Cox et al (2002)	Geodise	Condor					✓
30	Papazoglou (2003)	SOA	SGB (service grid bus)				✓	✓

Table 2: Classification based on service-orientation and middleware

[2] The big tick in Table 2 indicates that the paper places much emphasis on the application area(s) and the small tick indicates less emphasis.

2.2.2. Computational middleware

As the original intention of the Grid which is computational power, is almost accomplished with the metacomputing facilities, other problems began to emerge. The most urgent and significant was the need for a middleware that could allow interoperability among the diverse heterogeneous software and hardware located at the different supercomputing centre that made up the metacomputing. Not only interoperability, but the coordination and management of the diverse resources for optimum utilisation by the users at the different centre were also a problem. For example in most of the early Grid projects, just as storage became a central issue when the CPU cycle time efficiency was achieved, data management was an issue after achieving data storage (Gellrich et al., 2006). It is a progressive chain of challenges as the research will reveal the evolution of Grid computing over the years. Users hoping to parallelise a large single chunk of computational and data intensive job over several Grid nodes that have different platforms and run different software applications might get discouraged due to the inherent problems of lack of middleware (Goux et al., 2001). Present Grid platforms for EDO need to address the need to have a high-level application programmer interface (API) for Grid application programming. Goux et al. (2001) have developed such API called the Master Worker (MW). MW is a tool that allows users to easily distribute large-scale scientific computations on computational Grid. MW handles middleware issues using Condor as well as providing API for implementing master-worker algorithms with computational resources for EDO. Figure 9 shows some selected research papers against various middleware used for engineering design optimisation. Globus middleware occurred most in the research papers demonstrating its success among researchers.

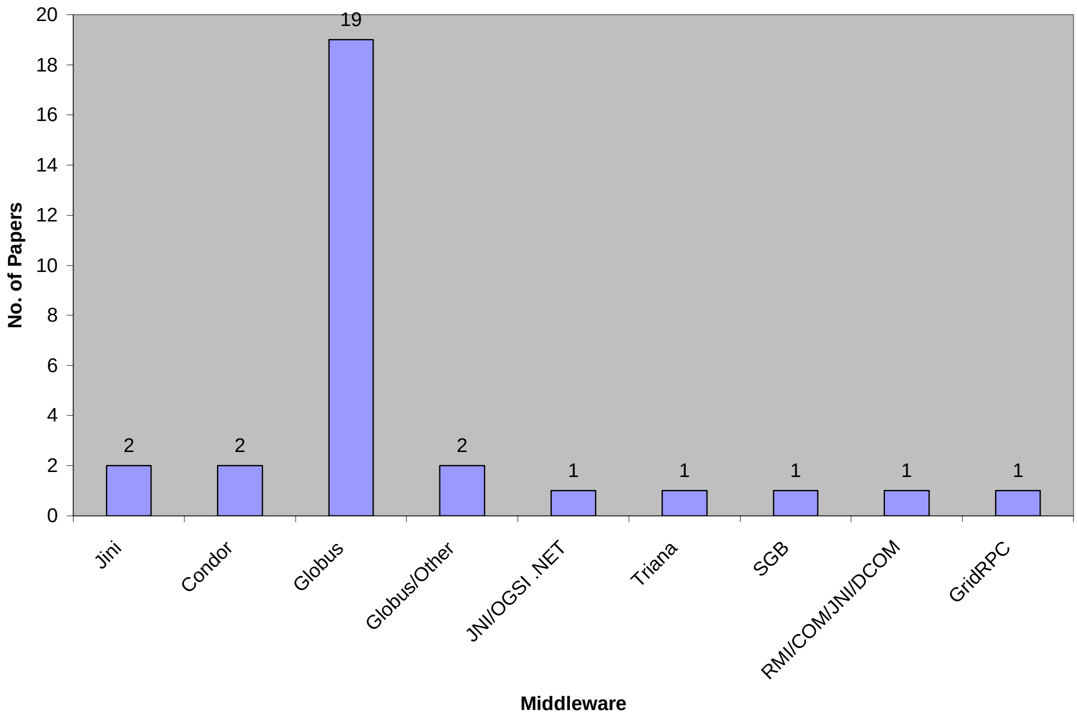

Figure 9: Middleware and numbers of papers

2.2.3. Data middleware

There is the need for special purpose middleware to cater for distributed data within a dynamic Grid environment. Different formats of databases are found in Data Grid. Middleware has the capacity to integrate virtual data and reuse for collaborative engineering design optimisation. Virtual data mechanisms and middleware enable the declarative specification of the recipes used to derive data such that transparent requests for products can be mapped into heterogeneous computation and data across multiple Grid computing and storage locations (Zhao et al, 2004). This is the motivation for future data manipulation over the Grid. Figure 10 is a typical arrangement of how virtual data is managed and shows the lifecycle of distributed Grid computing virtual data.

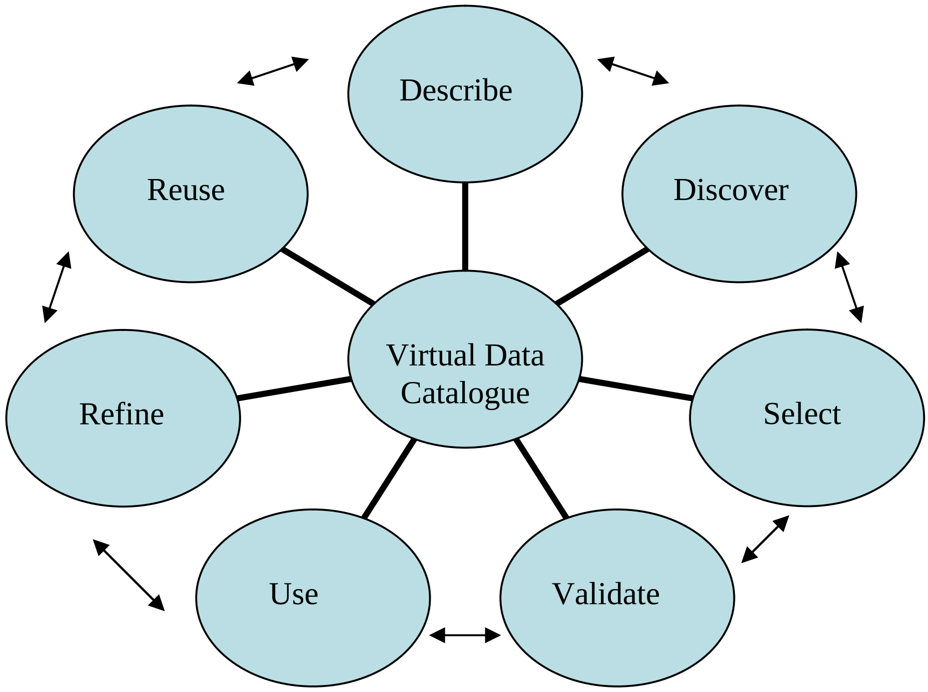

Figure 10: Lifecycle of virtual data within grid environment

2.2.4. Visualisation grid

Another challenge the first generation Grid-enabled infrastructures faced was the need to obtain real-time results and visualise them to acquire better understanding for decision making. The computational steering of parameters at run time for engineering design optimisation as the designer visualises the effect is very important. This allows other collaborators in the design process to join (Goodyer et al., 2005) in the simulation and to interact with the design system through both visualisation and steering from other sites of the Grid nodes. The common thinking among the users is that computing power and resources are needed away from local needs and that applications need to use distributed resources (Goodyer et al., 2005). Goodyer et al. (2005) maintained that Grid applications which are computationally intensive and collaborative in terms of the scientific community lead to two important questions. They are (1) how can knowledge and insight be acquired quickly from Grid application that runs on the distributed nodes? (2) How can these results and knowledge be shared among geographically dispersed scientists who might have different backgrounds and expertise? These questions obviously lead to the modification and enhancement of the Grid. To answer these questions, knowledge, collaboration and visualisation capabilities need to be part and parcel of the Grid. Engineering design optimisation (EDO) is a multidisciplinary activity which brings together various experts to yield improved designs. Because of the enormous data that can be generated during EDO, the conventional 2-D and even 3-D visualisation techniques are not adequate for decision making by engineers. Visualisation approaches such as Virtual Reality and Scientific Visualisation, Computational Steering concepts, Real Time Visualisation methods, and Cloud Visualisation methods (Eddy and Lewis, 2002) allow multidimensional visualisation for multi-objective optimisation processes. The recent advances in computer visualisation and Grid-enabled Virtual Reality (VR) allow designers to interact and manipulate vast amounts of data and metadata to further improve their efficiencies in producing competitive engineering products at affordable costs. Distributed Grid visualisation solves a great deal of resource discovery and allocation problems including the location of processing data (Brodlie et al., 2004). Figure 11 shows a simple Grid visualisation process for engineering design optimisation.

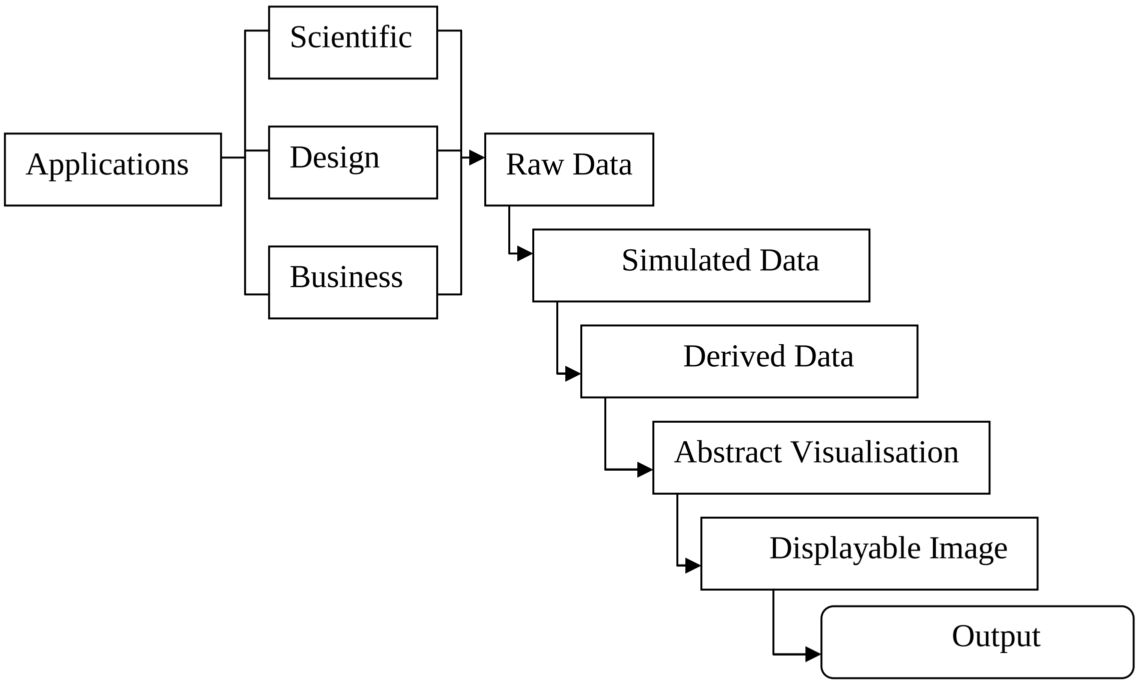

Figure 11: Grid visualisation process

2.2.5. Knowledge grid

The sequence of the Grid evolution continues with each successful or near successful implementation of phases. Successive phases lead to other phases due to challenges encountered. Having realised the importance of visualisation among collaborative Grid users, there is the need for these collaborators to share and reuse knowledge. A multidisciplinary field such as engineering design optimisation requires that each domain expert share his or her expertise with others so that there will be complete learning cycle for more efficient design process. It is not enough to have computational, data and visualisation capabilities. Though visualisation helps in uncovering some salient details of data, combining visualisation with knowledge gives a richer understanding of data as well as ensures reuse for computational steering. A vast amount of information is generated and stored in digital repositories during engineering design optimisation, yet it is often difficult to understand the important and useful information in those massive datasets (Cannataro and Talia, 2003a). The databases holding engineering design information need knowledge discovery agents to guide new optimisation processes. In this way, design cycle time will be reduced as well as improving the efficiency of the design engineers. Cannataro and Talia (2003a) described the use of knowledge discovery in databases (KDD) process for knowledge reuse in Grid computing. Cannataro and Talia (2003b) used KDD to demonstrate how data mining can be carried out using data classification, data clustering, events and values prediction, association rules discovery, and episode detection in Grid environment to achieve good knowledge management of large amount of data in distributed locations. Artificial intelligence (AI) techniques such as neural networks, fuzzy logic, and genetic algorithms are becoming important in complex optimisation processes within knowledge Grid environments. It is believed that knowledge Grid consists of resource space model (RSM) and operable knowledge browser (Zhuge and Liu, 2003). Figure 12 shows a typical knowledge space that can be implemented within Grid portal. The Grid portal conforms to the Globus single sign-on proxy encryption security feature. This feature enables grid users to access computational resources from multiple distributed sources with only single login credential that is performed once. This is important as it reduces the duplication and complication of multiple login credentials.

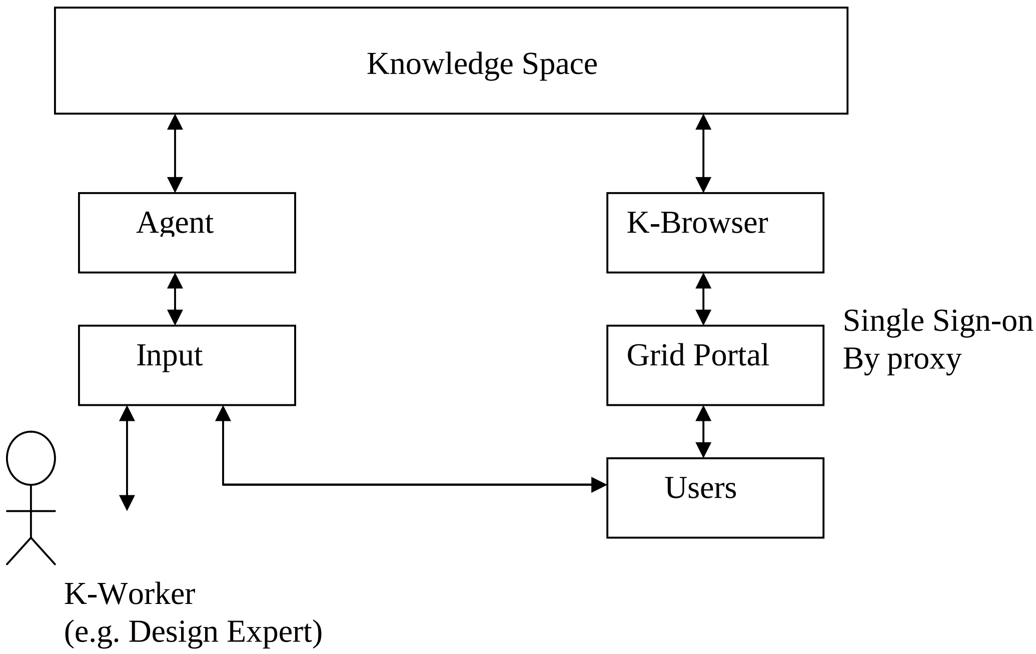

Figure 12: Knowledge Input and Creation space within Grid Portal

2.2.6. Examples of EDO applications in second generation grid

The second generation grid witnessed projects that are directly targeted at solving engineering design optimisation problems. One of the popular projects is the GEODISE (Grid-Enabled Optimisation Design Search for Engineering) project. Geodise is a UK e-Science project at University of Southampton in collaboration with some industries. The aim of the project is to provide a grid-enabled platform for engineers to collaborate and have easy access to optimisation tools from distributed locations. Other EDO projects include FIPER (Federated Intelligent Product Environment) and DAME (Distributed Aircraft Maintenance Environment) which aim at providing knowledge-based environment for engineers to perform optimisation activities in a more intelligent manner. DAME for example may be tested by Rolls-Royce in the A380 engine soon Ong et al. (2005).

2.3. Third generation of grid computing

Third generation of Grid computing addresses the need for service-oriented architecture and problem solving environments.

2.3.1. Web services

To understand and appreciate the concept of Grid services as part of the third generation of grid computing, it is important to trace its origin. The concept is an extension of the successes recorded in Web Services implementation and usage in e-commerce and e-business. For the Grid to gain world wide acceptance, it must play all these roles as well as many more as the Grid promises to involve more complex virtual users than the web has ever witnessed. It is instructive to note that conventional web services are not part of third generation of grid computing, but offer the idea towards Grid Open Services Architecture as a paradigm shift in the conventional computational Grid architecture. Figure 13 shows Web services architecture. Grid services architecture is based on the same concept.

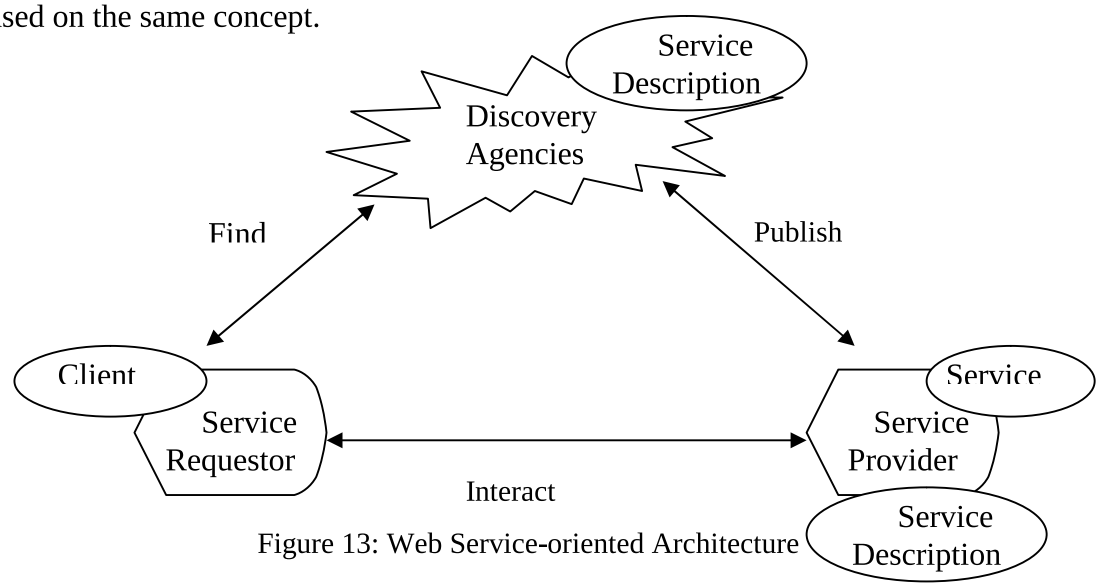

Figure 13: Web Service-oriented Architecture

A web service is an autonomous unit of application logic that provides either some business functionality or information to other applications through the internet connection (Hung and Li, 2001). It is based on XML standards and protocols such as simple object access protocol (SOAP), web services description language (WSDL), and universal description, discovery and integration (UDDI) services. Just like any service, web services have service providers and service requestors. Web service providers publish resources that are needed by service requestors. The service requestors pay for the services they use. Today's pricing needs of web services are static but more complex dynamic pricing policies are needed for the use of computing or data in future web or Grid services. In this case the model will address issues such as quality of service (QoS) provided by web services providers and cost of service (CoS) paid by requestors (Hung and Li, 2001). The success of web services is partly due to its ability to resolve interoperability problems. Web services resources are obtained by requestors regardless of the different operating systems, platforms, browsers, network topology, or programming languages used (Nandigam et al, 2005). Before the advent of web services, engineering design optimisation was done on individual engineer basis. With web services, the whole process of engineering design optimisation encompassing the multidisciplinary disciplines involved (Woyak et al., 2004).

2.3.2. Service-oriented grid computing

Like all infrastructures, users want to see how Grid computing can be applied directly to solve their daily needs. For example, infrastructures such as electricity, railroads, and telephone were successful because of the visible service they provided at their inception as services. Users will naturally endorse infrastructures when the benefits are practically visible in the way they do things. This perhaps informed the decision of researchers to develop a Grid utility model. Grid applications should be developed to compete with windows and web based applications to deliver real benefits that have comparative economic advantage over existing applications. The third generation of Grid computing research papers concentrated efforts on providing Grid platforms for Grid application developers. The web made significant impact not because it runs on the internet but because it provides an electronic market place for doing commerce and business. Problem solving environments such as FIPER (Federated Intelligent Product Environment), SORCER (Service-Oriented Concurrent Environment) and GEODISE (Grid-Enabled Optimisation Design Search for Engineering) were developed in line with this vision. Figure 14 shows the number of papers that researched on making the Grid a utility platform for service-oriented computing and advanced middleware services. There is an increase in research papers in service-oriented Grid-enabled computing from 2000 to 2004. The third generation Grid research started combining service-oriented models with problem solving environments from 2005. This is why engineering design optimisation as a process in engineering discipline received attention from Geodise, FIPER and SORCER problem solving environments. This is part of the efforts researchers are making to showcase that Grid-enabled service-oriented problem solving environments can solve real and practical daily problems in science, engineering, and business.

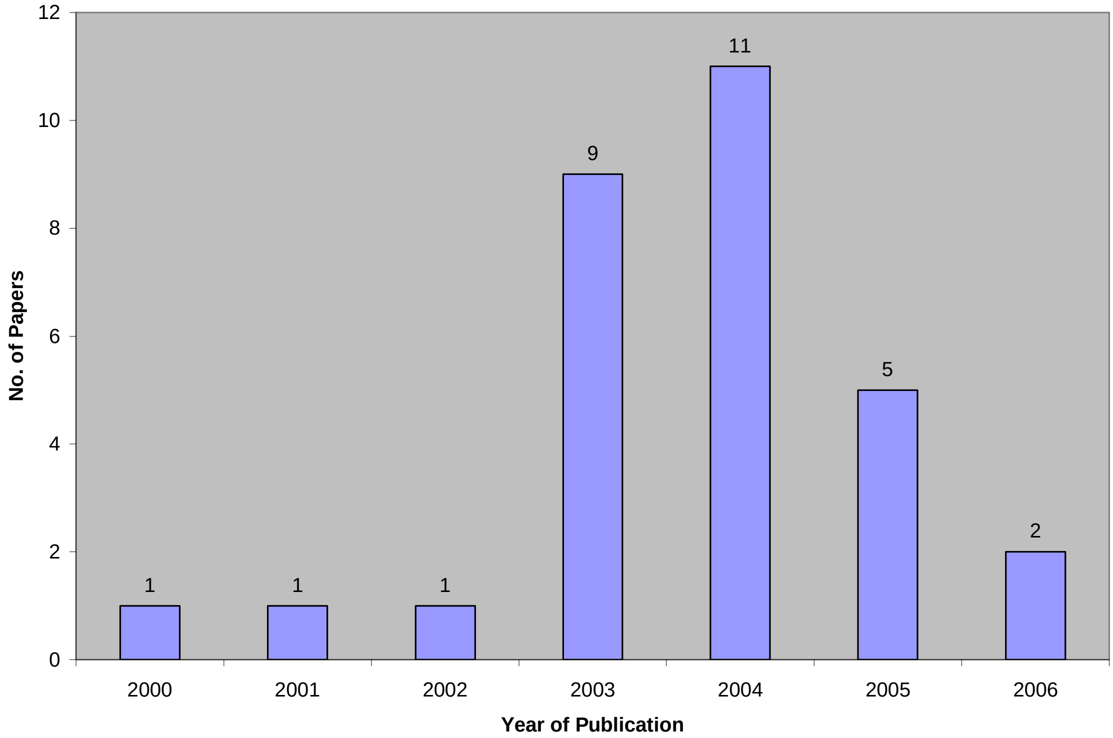

Figure 14: Service-orientation model papers and years of publication

Figure 15 shows the number of research papers that worked on different service-oriented models with regard to engineering design optimisation. Geodise received the highest attention from the papers collected followed by OGSA. The Grid model allows users to use applications such as Matlab per usage instead of per license. That is users invoke Matlab executable as a service a pay for the time they use it instead of buying the software.

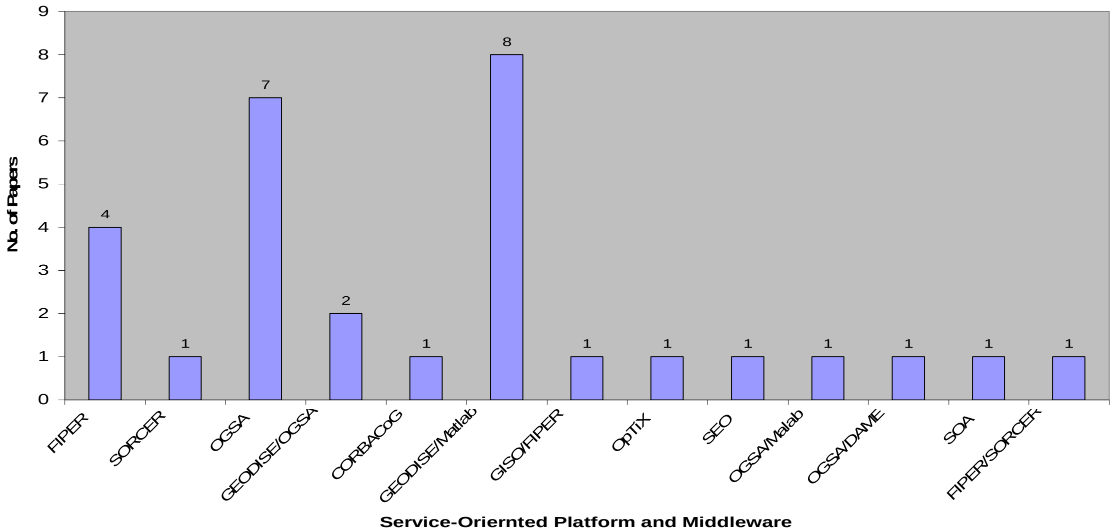

Figure 15: Service-oriented models and number of papers

The success of Web services is a booster to the concept of Grid services. The innovative development of Web Services Resource Framework (WSRF) that allows Web services and Grid services to converge and work on the same platform and interact simultaneously has transformed positively the vision of service-oriented Grid computing. Collaboration is at the heart of Grid computing (Song et al., 2004a). A service-oriented Grid-enabled web-based computing has the robust capabilities to enhance collaboration in an extended manner that has never been possible. For example, in engineering design optimisation where computing cycle, data storage, optimisation codes, multidisciplinary knowledge, and expertise are distributed across geographical boundaries around the world, cooperative organisations can agree on service (supply and demand) policies and register on a Grid in which participant published resources are available for usage by all. In this way, users do not need to own all resources for design optimisation process. In complex engineering design problems, analysis models are used which require computational structural mechanics (CSM) and computational fluid dynamics (CFD) simulations taking hours or even weeks of computing cycle time (Song et al., 2004b). Song et al. (2004b) suggested that a service-oriented Grid has the capability of globally improving the design and overcoming the computational and data handling requirements with stochastic techniques such as evolutionary algorithms. Figure 16 is the process of Web and Grid convergence in technology and application areas. The convergence point is facilitated by the Web Services Resource Framework (WSRF). The convergence shows trend from I-WAY (Information Wide Area Year), Globus versions (GT1 to GT3) to OGSA (Open Grid Services Architecture) and GSDL (Grid Services Description Language) on the Grid site. On the Web convergence site, the convergence begins from HTTP to WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol) and UDDI (Universal Description, Discovery and Integration). For the service-oriented Grid to mature fully, participants (producers and consumers of services) need to have trust in the system. The security requirements of a dynamic and coordinated Grid are enormous. This perhaps is the reason for the complex Globus Security Infrastructure (Goel and Sobolewski, 2003) proposed for service-based Grid secured management.

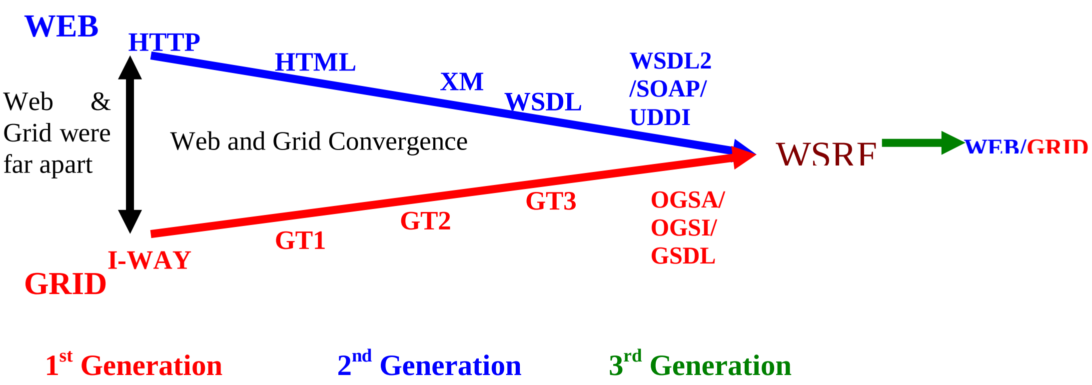

Figure 16: Web and grid convergence process as the generations of grid mature

Since service-oriented computing is all about creating a market place for supply and demand playing ground for computing resources, developing a robust accounting system that can dynamically charge consumers of resources according to quantity and/or time of usage as well as remits same amount to suppliers account is important. The accounting feature needs to take into consideration many economic issues. Quality of service (QoS) might be relative to various providers and users of resources and so different compromising metrics will be used to harmonise charge rates. Barmouta and Buyya (2003) proposed a Grid accounting system called GridBank. GridBank architecture consists of several components for performing different tasks to accomplish a robust professional accounting scenario for service computing.

2.3.3. Semantic web

Semantic Web is an extension of the Web. Semantic Web is centred on metadata, ontologies, inference rules and software agents (Nandigam et al., 2005). In this context, information is given well-defined meaning, better enabling computers and people to work in cooperative manner (Cannataro and Talia, 2004). By defining and linking data intelligently, effective automation, discovery, integration and reuse across applications and services are enhanced. The Web is meant for human interpretation while the Semantic Web is meant for Web resources to be machine-understandable semantics. In a nutshell, Semantic Web is information expressed in special machine-targeted language, whereas the Web is information targeted at human consumption expressed in various natural languages. The multidisciplinary and knowledge driven area such as engineering design optimisation requires semantic and ontology for managing and reuse of resources. The Web services and Grid services discussed earlier need enhancement. The ultimate purpose of the Grid is to support collaboration and knowledge discovery (Goble and Roure, 2002).

2.3.4. Semantic grid

Semantic Grid incorporates ontologies, knowledge, agent-based applications and inference rules in managing heterogeneous and dynamic resources with high level of coordination of Grid services. The Semantic Grid seeks to incorporate the Semantic Web into its implementation (Cannataro and Talia, 2004). The next generation of Grid services stands to benefit from Semantic Grid. High level services such as workflow generation and systematic process flow as witnessed in engineering design optimisation requires the use of intelligent agent-based optimisation environment to aid in search exploration (Gil et al., 2004).

What will complement the present efforts in open Grid services framework is the syntactical descriptions of service and application relationships. The interfaces of interacting services should be able to discover, reconfigure and adapt to services provision and service request autonomously through the implementation of semantic Grid Ontologies (Hau et al, 2003). Protocols such as OWL (Web Ontology Language), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration), and XML are essential for identifying services based on semantic expressed capabilities in relation to some ontology concepts for discovery and adaptation of dynamic Grid services management. Organising the large amount of data produced and analysing it is a challenging task. The emergence of semantic data Grid

(SDG) is seen as the solution to this problem (Schwidder et al., 2005). The effective automated discovery, analysis and integration of data and metadata in Grid environments can only be achieved through some kind of semantic annotation of process flow (Fox, 2003) in engineering design optimisation. The future success of Grid computing depends on the availability of semantic/knowledge-rich resources for science and business (Chen et al., 2004). The Semantic Grid is the Internet-centred interconnection environment that can effectively organise, share, cluster, fuse, and manage globally distributed versatile resources based on the interconnection semantics (Zhuge, 2005) coordinating the activities of design engineers. The middleware solves the original problem of heterogeneity so that computational resources and services can work together in cooperation; a new problem that needs to be addressed by Semantic Grid is interoperability across time as well as space for reuse of services, information and knowledge (Roure et al, 2005)

2.3.5. Examples of EDO applications in third generation grid

Engineering design optimisation received attention in third generation of grid computing in the areas of service provision and problem solving environments. Projects such as SORCER (Service-Oriented Concurrent Environment), GRIA (Grid Resources for Industrial Applications) and Globus toolkit version 4 (GT4) are aimed at providing service-oriented platforms for grid-enabled engineering design optimisation. GRIA for example is used by BAE Systems as a grid communication protocol within its web services system.

3. Future trends in grid computing for engineering design optimisation

There is a lot to be seen in the future of Grid computing. There are many unanswered research questions. Besides, Grid computing has not yet started yielding the expected benefits for end users. The future research issues will concentrate on extending ideas and implementation of autonomic computing, economic models and ubiquitous computing.

3.1. Autonomic grid computing

Future Grid computing resources and services should be self-adjustable, self-adaptable, self-manageable and self-configurable. Unlike conventional information systems that have few interactions, the Grid will have multitude of users interacting and sharing heterogeneous resources dynamically. This presents a great deal of management issues which calls for the resources and services to automatically manage themselves without the intervention of a systems administrator. For example, complex engineering design optimisation in the future will be done within a problem solving environment that has resources located at different centre of expertise and optimisation code to be used will be selected based on the constraints, inputs and parameters of the problem presented without the intervention of the design engineer (Parmee et al., 2005). The interfaces of services and resources configure and adjust themselves automatically. In this way, the efficiency of design experts will improve tremendously. Another example is in the optimisation of oil reservoir placement. The location of wells in oil and environmental applications significantly affect the productivity and economic benefits of surface reservoir (Parashar et al., 2005). To get the best benefit, the determination of optimal location is important and should not be risked through manual processes. This calls for autonomic Grid computing services that will use intelligent reservoir simulators and select the best optimisation search for the location of an economic oil location. This grand challenge of systems managing themselves requires specific and generic autonomic computing elements (Kepart, 2005).

3.2. Ubiquitous grid computing

Ubicomp (ubiquitous computing) is a grand challenge vision which allows people and environments enabled with computational resources provide information and services when and where desired (Hightower and Borriello, 2001) for pervasive and distributed coordination and management of cooperative organisations. Grid-enabled ubiquitous computing will lead to systems knowing the physical location of things so that they can record them and report them to those that need to know. For example, the system could help the search and rescue team locate exactly the position of victims and their condition for quick evacuation. This system will revolutionised the area of mobile computing for efficient identification of agents. This grand challenge area will enable computers to be aware of our environment and take critical actions. The pervasive nature of Grid computing coupled with the multidisciplinary and distributed nature of engineering design optimisation require the use of ubiquitous Grids in the future.

3.3. Service economic grid model for grid computing

The Grid needs to address issues concerning service on-demand basis and charge per usage for end users. A dynamic metering and accounting system for dynamic usage of resources and services will be more appropriate for Grid computing service-oriented architecture. Presently, Web services use static charging system for e-commerce. For example, buying a book from amazon.com is charged per the price of the book. The accounting prizing system in Grid will be based on many criteria such as quality of service, policies, time of usage and amount of usage and so on. The dynamic changes in services and resources will also affect the pricing system. Grid Resource Brokers (GRB) component of the middleware (Barmouta and Buyya, 2003) can be improved to handle dynamic Grid accounting process. This is different from the web services in which users pay for the book as well as the bandwidth for the internet service provider as Grid services can name, create and monitor different states of the sources of the services. Engineering design optimisation is multidisciplinary and dynamic in nature and as such it requires dynamic coordination of resources in an economic manner. The development of robust and secured economic and accounting models for the Grid is vital.

The computational Grid has enhanced the capability of distributed computing systems to model and simulate complex systems in the scientific, engineering and commercial domain (Darema, 2004). The future trend in engineering design optimisation using Grid-enabled technologies will be based on ubiquitous (pervasive) and on-demand service-oriented access to optimisation resources. The notion of dynamic data driven application systems (DDDAS) would see the Grid encapsulating services such as quality of data (QoD) or quality of service (QoS) assessment, uncertainty model services, and sensor services in EDO process. According to Lloyd (2003), the electricity Grid from which the computational grid was conceived, had an equally visionary focus, with more precise focus on the needs of its immediate market than the computational Grid of today. Lloyd argued that the electric power Grid had a well defined set of users that the expected net income would produce a viable investment in protecting the intellectual property associated with its invention and provision as a utility as well as return on investment (ROI) to investors so that more capital could be raised to make the “Grid” available to more users in a pervasive way. For the computing grid to assume a global infrastructural status like the electric power Grid, it must have a clearly defined service oriented nature with economically active users who can confidently pinpoint and identify associated value or “utility” to that service which should ordinarily exceed value/cost ratio of other existing alternative technologies such as the internet services and should also cover cost of migration from those existing technologies.

In the days of Edison (considered as the father of electric power grid), there was an existing utility (gas) and it was a matter of getting around the competitive economic advantage over the gas lighting as a benchmark and substitute it with electric lighting. Edison also paid close attention to the needs of various customers and had various sections of his company deal with different products relating to specific customers. For example, Edison Lamp Works produced lamps, Edison Illuminating Company supplied lighting, Edison Machine Works built dynamos and Edison Electric Tube Company constructed underground conductors. This organised level of division of service provided a better focus and specialisation that was used as a productivity advantage over other existing competing utilities such as gas lamp. Today’s computational Grid is also drifting in the same direction. The initial conception of the Grid was mostly for scientific computation and data intensive applications. The present shift in emphasis to a more service oriented Grid informed the decision to have different types of Grids. There are now computational Grids, data Grids, enterprise Grids, extraprise Grids, global Grids, and many more. Just like in Edison’s companies which serve different customers and different purposes, these different grids serve different users and different purposes. For example, the computational Grid serves users who need computing power for their jobs, data Grid serves users who analyse data and use data intensive applications, extraprise Grids are for companies and businesses to collaborate with their customers and partners, and global Grids will be a service utility for general purpose business such as in e-commerce. The only way to keep the tempo of enthusiasm in Grid computing infrastructure is to ensure that its business value, reliability and application performance outweighs today’s competing technologies such as parallel distributed technology and Web services technology. These values and applications must be visible to the “real” end-users of Grid technologies. For example, services should be categorised based on private, commercial or industrial usage. The charge rates should also be based on the same categorisation. In electricity charge rates, commercial and industrial users have more charge rates than private users. Furthermore, services for engineering design optimisation should be based on simple, medium or complex optimisation problems. This categorisation will serve as an incentive to many users.

3.4. Grid computing security

The only way the concept of Grid services will record huge support is when users are assured of the safety of their resources such as information and knowledge from their competitors. The Open Grid Services Architecture (OGSA) and the middleware must address issues of incorporating legacy resources without compromising security. One major factor that will attract open service-oriented adaptation of the Grid is security assurance to users. For the Grid to be wholly embraced as a "trusted" utility, the security feature must perform better than existing conventional public key encryption infrastructure that provide security for most networks and distributed computing facilities. Research in the area of security resiliency shows that it is important to have good fault tolerant capabilities in distributed systems even in the presence of external (Zhonhua et al., 2003) attacks and intrusions. This is necessary for distributed systems to deliver essential services even in the face of attack, failure, or any incident. Grid computing which is envisaged to be a service-oriented infrastructure by the Global Grid Forum (GGF) now Open Grid Forum (OGF). OGF with other researchers have devoted great effort in developing a secured grid security infrastructure (GSI) with many features that address these issues.

4. Summary and conclusions

Engineering design optimisation requires the participation of several experts. These experts are difficult to be found in one geographical location. The need to develop platforms for design experts to collaborate and work together is to overcome this distance-geographical barrier. In addition, Grid-based problem solving environments provide easy access to computational resources for design engineers working on different tasks concurrently. The service-oriented architecture allows resources for optimisation to be published by service providers and used by service requestors dynamically. This paper highlighted the trend of Grid computing. The major trends are classified as first, second and third generations of Grid computing. The first generation is concerned with computational and data intensive applications of Grid. This is accomplished by linking several supercomputing (metacomputing) stations to harness computing power. Example of grid project in the first generation is I-WAY project. The second generation provided middleware and communication protocols to solve problems of interoperability during implementations of metacomputing (first generation) test-beds. Examples of engineering design projects in the second grid generation include GEODISE and FIPER projects. The third generation of Grid computing shifted emphasis to service-oriented and problem solving environments. EDO projects in the third generation grid include SORCER and GRIA projects. The report considered engineering design optimisation (EDO) as a suitable application area for Grid computing. This is because EDO is computationally and data intensive in nature. EDO also involves the collaboration of multidisciplinary experts who may be located in different geographical regions. The computational and data capabilities of Grid computing coupled with its distributed nodes for collaboration makes Grid attractive for EDO. Knowledge reuse is another challenge in EDO. Semantic and Knowledge Grids have the potential to address knowledge reuse in EDO by capturing and storing EDO processes that can be retrieved and reused by new design engineers.

The future trend in Grid computing research is concerned with autonomic computing, ubiquitous computing and economic models. Autonomic computing allows self management of Grid resources and services as the system becomes complex. Ubiquitous computing enhances the pervasiveness of the Grid. Economic models provide ‘plug-and-use’ and pay per usage mechanisms for Grid services providers (owners or producers) and requestors (users or consumers). The contribution made by this report is the classification of grid evolution and proposed future trends in research. This provides state-of-the-art overview of grid computing.

5. Acknowledgements

The researchers wish to thank the Petroleum Technology Development Fund (PTDF), Nigeria and Cranfield University, UK for sponsoring the research.

References

Aloisio, G., Cafaro, M., Fiore, S. and Mirto, M. (2004). The GRelC Library: A Basic Piller in the Grid Relational Catalog Architecture. Proceedings of the International Conference on Information Technology (ITCC'04): IEEE Computer Society.

Antonioletti, M., Atkinson, M., Baxter, R., Borley, A., Hong, N. P., Colins, B., Hardman, N., Hume, A. C., Knox, A., Jackson, M., Krause, A., Laws, S., Magowan, J., Paton, N. P., Pearson, D., Sugden, T., Watson, P. and Westhead, M. (2005). The Design and Implementation of Grid Database Services in OGSA-DAI. Concurrency and Computation: Practice and Experience.

Antonioletti, M., Hong, N. C, Hume, A., Jackson, M., Krause, A., Nowell, J., Palansuriya, C., Sugden, T. and Westhead, M. (2003). Experiences of Designing and Implementing Grid Database Services in the OGSA-DAI Project. Designing and Implementing Grid Database Services Workshop, UK: GGF9.

Barmouta, A. and Buyya, R. (2003). GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration. Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03): IEEE Computer Society.

Berlich, R., Kunze, M. and Schwarz, K. (2005). Grid Computing in Europe: From Research to employment. Australian Workshop on Grid Computing and e-Research; Newcastle, Australia. Australia: Australian Computer Society, Inc.

Berman, F., Fox, G. C. and Hey, A. J. G. (2003). Grid Computing: Making the Global Infrastructure a Reality. USA: John Wiley & Sons Ltd.

Brodlie, K. W., Duce, D. A., Gallop, J. R., Walton, J. P. R. B. and Wood, J. D. (2004). Distributed and Collaborative Visualization. The Erographics Association and Blackwell Publishing Ltd., 23(2):223-251.

Brodlie, K., Duce, D., Gallop, J., Sagar, M., Walton, J. and Wood, J. (2004). Visualization in Grid Computing Environments. IEEE Visualization 2004; Austin, Texas. USA: IEEE: 155-162. ISBN: 0-78038788-0/04.

Cannataro, M. and Talia, D. (2003a). The Knowledge Grid: Designing, Building, and Implementing an Architecture for Distributed Knowledge Discovery. Communications of the ACM. 46(1):89-93.

Cannataro, M. and Talia, D. (2004). Semantics and Knowledge Grids: Building the Next-Generation Grid. IEEE Intelligent Systems. 56-63.

Cannataro, M. and Talia, D. (2003b). Towards the Next-Generation Grid: A Pervasive Environment for Knowledge-Based Computing. Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03)USA: IEEE Computer Society; ISBN: 0-7695-1916-4/03.

Cao, J., Jarvis, S. A., Spooner, D. P., Turner, J. D., Kerbyson, D. J. and Nudd, G. R. (2002). Performance Prediction Technology for Agent-based Resource Management in Grid Environments. Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02): IEEE.

Cao, Junwei; S., Daniel P., Jarvis, S. A., Saini, S. and Nudd, G. R. (2003). Agent-Based Grid Load Balancing Using Performance-Driven Task Scheduling. Proceedings of the International Parallel and Distributed Processing Symposium Symposium (IPDPS'03)USA: IEEE Computer Society; ISBN: 0-76951926-1/03.

Chen, L., Cox, S. J., Tao, F., Shadbolt, N. R., Puleston, C. and Goble, C. (2004). Empowering Resource Providers to Build the Semantic Grid. Proceedings of the IEEE/WIC/ACM Intenational Conference on Web Intelligence (WI'04): IEEE Computer Society.

Darema, F. (2005). Grid Computing and beyond: The Context of Dynamic Data Driven Applications Systems. Proceedings of the IEEE USA: IEE: 692-697.

Deelman, E., Kesselman, C., Mehta, G., Meshkat, L., Pearlman, L., Blackburn, K., Ehrens, P., Lazzarini, A., Williams, R. and Koranda, S. (2002). GriPhyN and LIGO, Building a virtual Data Grid for Gravitational Wave Scientists. Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC'02): IEEE Computer Society.

Deelman, E., Singh, G., Atkinson, M. P., Chervenak, A., Hong, N. P. C., Kesselman, C., Patil, S., Pearlman, L. and Su, M. (2004). Grid-Based Metadata Services. Procceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04): IEEE Computer Society.

Eddy, J. and Lewis, K. (2002). Multidimensional Design Visualization in Multiobjective Optimization. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; Atlanta, Georgia. USA: AIAA; ISBN: AIAA 2002-5621.

Eres, M. H., Pound, G. E., Jiao, Z., Wason, J. L., Xu, F., Keane, A. J. and Cox, S. J. (2003). Implementation and Utilisation of a Grid-Enabled Problem Solving Environment in Matlab: Elsevier.

Foster, I., Insley, J., Laszewski, G., Kesselman, C. and Thiebaux, M. (1999). Distance Visualization: Data Exploration on the Grid. Scintific VisualizationUSA: IEEE; ISBN: 0018-9162/99.

Foster, I. and Kesselman, C. (1999). The Grid: Blueprint for a New Computing Infrastructure. San Francisco, Carlifornia, USA: Morgan Kaufmann Publishers, Inc.

Foster, I., Kesselman, C., Nick, J. M. and Tuecke, S. (2002a). Grid Services for Distributed System Integration. IEEE. 37-46.

Foster, I., Vockler, J., Wilde, M. and Zhao, Y. Chimera: (2002b). A Virtual Data System for Representing, Querying, and Automating Data Derivation. Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM'02): IEEE Computer Society, ISBN: 1099-3371/02.

Fox, G. (2003). Data and Metadata on the Semantic Grid. Web Computing: IEEE Computer Society and the AIP; Computing in Science and Engineering.

Gellrich, A., Ernst, M., Fuhrmann, P., Gulzow, V. and Riese, M. (2005). Grid Technology in Production at DESY. Nuclear Instruments and Methods in Physics Research. 559:31-34.

Gil, Y., Deelman, E., Blythe, J., Kesselman, C. and Tangmunarunkit, H. (2004). Artificial Intelligence and Grids: Workflow Planning and Beyond. IEEE Intelligent Systems. 26-33.
Goble, C. and Roure, D. (2002). The Grid: An Application of the Semantic Web. SIGMOND Record. 31(4):65-70.

Goel, S. and Sobolewski, M. Trust and Security in Enterprise Grid Computing Environment. Proceedings of the IASTED International Conference on Communication, Network and Information Security; New York. 2003.

Goodyer, C. E., Berzins, M., Jimack, P. K. and Scales, L. E. (2005). A Grid-Enabled Problem Solving Environment for Parallel Computational Engineering Design. Advances in Engineering Software. 37:439449.
Goux, J., Kulkarni, S., Yoder, M. and Linderoth, J. Master-Worker: (2001). An Enabling Framework for Applications on the Computational Grid. Cluster Computing. 4:63-70.

Hau, J., Lee, W. and Newhouse, S. (2003). Autonomous Service Adaptation in ICENI using Ontological Annotation. Proceedings of the Fourth International Workshop on Grid Computing (GRID'03): IEEE

Computer Society.

Hightower, J. and Borriello, G. (2001). Location Systems for Ubiquitous Computing. IEEE Location Aware Computing. 57-66.

Hung, P. C. K. and Li, H. (2001). Web Services Discovery Based on the Trade-off between Quality and Cost of Service: A Token-based Approach: ACM. ISBN: 1073-0516/0300-0034.

Nandigam, J., Gudivada, V. N. and Kalavala, M. (2005). Semantic Web Services. CCSC: Midwestern Conference: JCSC.

Parashar, M., Matossian, V., Bangerth, W., Klie, H., Rott, B., Kurc, T., Catalyurek, U. J., and Wheeler, M. F. (2005). Towards Dynamic Data-Driven Optimization of Oil Well Placement; Springer-Verlag Berlin Heidelberg. ICCS: 656-663.

Parmee, I. C., Abraham, J., Shackelford, M., Rana, O. F. and Shaikhali, A. (2005). Towards Autonomus Evolutionary Design Systems via Grid-Based Technologies. Proceedings of ASCE 2005 International Conference on Computing in Civil Engineering; Cancum, Mexico. Mexico: ASCE.

Rajasekar, A., Wan, M. and Moore, R. (2002). MySRB & SRB - Components of a Data Grid. Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC'02): IEEE Computer Society.

Rajasekar, A., Wan, M., Moore, R., Kremenek, G. and Guptil, T. (2003). Data Grids, Collections, and Grid Bricks. Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems Technologies (MSS'03): IEEE Computer Society; ISBN: 0-7695-1914-8/03.

Roure, D., Jennings, N. R. and Shadbolt, N. R. (2005). The Semantic Grid: Past, Present, and Future. Proceedings of The IEEE. 93(3):669-681.

Schwidder, J., Talbott, T. and Myers, J. (2005). Bootstapping to a Semantic Grid. 2005 IEEE International Symposium on Cluster Computing and the Grid: IEEE.

Song, W., Keane, A. and Cox, S. (2004b). CFD-Based Shape Optimisation with Grid-Enabled Design Search Toolkits. Proceedings of UK e-Science All Hands Meeting 2003, UK: 616-626.

Song, W., Ong, Y. S., Ng, H. K., Keane, A., Cox, S. and Lee, B. S. (2004a). A Service-Oriented Approach for Aerodynamic Shape Optimisation across Institutional Boundaries. 8th International Conference on Control, Automation, Robotics and vision; Kunmlng, China.
Venugopal, S., Buyya, R. and Winton, L. (2004b). A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids. 2nd Workshop on Middleware in Grid Computing; Toronto, Canada. Canada: ACM, ISBN: 1-58113-950-0.

Wan, M., Rajasekar, A., Moore, R. and Andrews, P. (2003). A Simple Mass Storage System for the SRB Data Grid. Proceedings of the 20th IEEE /11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS'03): IEEE Computer Society; 2003ISBN: 0-7695-1914-8/03.

Wason, J., Jiao, Z., Song, W., Price, A. and Cox, S. (2004). Deployment and Exploitation of Grid-enabled Data Management for Engineers. UK e-Science AHM 2004UK: UK e-Science AHM.

Woyak, S., Kim, H., Mullins, J. and Sobieszczanski-Sobieski, J. (2004). A Web Centric Architecture for Deploying Multi-Disciplinary Engineering Design Processes. 10th AIAA/ISSMO Multidisciplinary

Analysis and Optimization Conference; Albany, New York. USA: AIAA Inc.

Zhao, Y., Wilde, M., Foster, I., Voeckler, J., Jordan, T., Quigg, E. and Dobson, J. (2004). Grid Middleware Services for Virtual Data Discovery, Composition, and Integration. 2nd Workshop on middleware for Grid Computing; Toronto, Canada. Canada: ACM.

Zhuge, H. (2003). China's E-Science Knowledge Grid Environment. IEEE Intelligent Systems. 13-17.

Zhuge, H. and Liu, J. (2005). A Fuzzy Collaborative Assessment Approach for Knowledge Grid. Future Generation Computer Systems. 20:101-111.

