

Collaboration and Community Grids

Geoffrey Fox

*Community Grids Laboratory, Computer Science Department and School of Informatics
Indiana University Bloomington
gcf@indiana.edu*

ABSTRACT

We study Grid architectures and relate them to the ideas of collaboration and community networks. We discuss services and architectural principles that support both synchronous and asynchronous collaboration and the integration with community networks and peer-to-peer systems. We identify 18 core features and services that characterize Grid systems. We conclude with a proposed Semantic Scholars' Grid integrating these concepts to support information discovery and sharing in communities of users.

KEYWORDS: Grids, Collaboration, Community, Services, P2P, Web Services

1. INTRODUCTION

The concept of Collaboration is intuitively obvious but nevertheless not used in consistent fashion by different communities. For example some applications specialize collaboration to either its synchronous or asynchronous form without noting this. Further collaborative systems are often built but not described in these terms. Here we note many community networks and peer-to-peer systems which clearly have significant collaborative properties. In this paper, we try to use a broad net to catch many of the different approaches to collaboration and discuss and illustrate Grid approaches to all of them. The major activities we will highlight in this paper are

1. The traditional Grid problem [1,2] of the sharing of large data and compute resources and the support of international scientific and engineering research teams (e-Science)
2. The Net Centric Environment (NCE) or Net Centric Operations and Warfare (NCOW) target architecture developed for the next generation Department of Defense systems [3-5]
3. The community and social networks exemplified by systems like <http://del.icio.us>. Wikis, Blogs and the search-enriched Internet

4. Peer-to-peer systems including the related chat systems and forums.
5. e-learning allowing distributed education and more generally collaboratories and portals supporting groups in other areas including software development (the notorious outsourcing) and other global business activities.

In section 2, we review and synthesize Grids with a service oriented architecture that captures the experience and practice of the major international Grid projects. This is summarized in the final Table 4 and exemplified in Figs. 4 and 5. This is followed by a discussion in Section 3 of different styles of collaboration and their characteristics. Then section 4 explains how one forms Collaboration and Community Grids based on these general ideas or more precisely on general services.

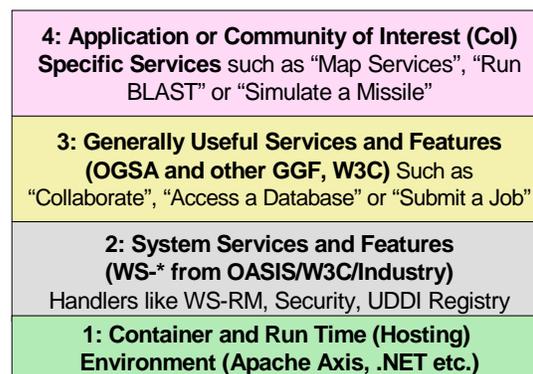


Figure 1. The Grid and Web service Institutional Hierarchy

2. GRID AND WEB SERVICES

Web Service-based SOA systems [6] are built on XML-based service description languages (WSDL) and message formats (SOAP). A review of Web Service concepts is given in [7]. In Fig. 1, we illustrate the current institutional hierarchy of Grid services. We call it "institutional" as the four blocks of service define groups

of services or service support of increasing specialization as we move up the figure. In an SOA, we are building services, their interactions (namely messages) and the support for the two fundamental concepts of messages and services. At the bottom level we have what is usually called the hosting environment, which forms the virtual machine on which we are building the “distributed service operating system” contained in the next layer. For services constructed from Java, Apache Axis is the usual container and it provides the message processing needed by the multiple services in the container.

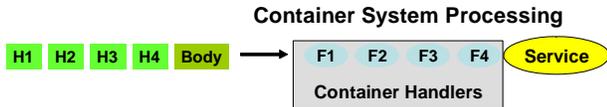


Figure 2. Message Processing in a Container

As shown in Fig. 2, a SOAP message contains multiple headers and a body. The headers are processed by handlers controlled by the container and these capabilities are included in the second level of Fig. 1. They include operations such as security, service addressing, routing, reliability and possibly conveying aspects of state and meta-data. One can consider handlers as the core system services for Web services. The handlers will in general modify the SOAP envelop contents, which are then formatted by the container so that it can be processed by the appropriate service instance. This instance could be implemented as a Java method corresponding to the WSDL-specified XML in the body. Microsoft’s .NET provides similar capabilities for Windows environments. Note that the SOAP messages can be transported by any mechanism for which a binding can be defined. This includes the normal HTTP transport but also may be message-oriented middleware to give environments like those in modern enterprise software environments such as those using IBM’s MQSeries Messaging or Java Messaging Service.

The capabilities described above are needed in all aspects of Web and Grid service implementations and several major international activities aim at setting the standards for the service and handler interfaces. The core level 1 and 2 specifications of Fig. 1 are often called the WS-*. In Table 2, we list the broad areas covered by this process which involves multiple standards agencies (OASIS, W3C, the Global Grid Forum, Distributed Management Task Force) and companies such as IBM and Microsoft working inside and outside the community bodies and in different combinations. There are over 60 WS-* proposal specifications – mostly initiated in the last 4 years – with a coverage indicated in Table 1. Column 1 lists our classification of the area and a very incomplete sample of the proposed specifications for each area are given in column 2. See appendices in [1] for more information.

Note that all handler specifications are given in one or more areas of Table 1. However this table also includes system services which are broad in scope and so fit in level 2 but are not processed in handlers. Areas 4 and 10 of Table 1 correspond to workflow and user interfaces – core capabilities but not associated with handlers. This is clarified in Fig. 3 which has a functional Web service hierarchy with pervasive system services like security (termed A in Fig. 3) separated from workflow (in Fig. 3(E) labeled manipulating and linking services) and portals in boxes F and G in Fig. 3.

Table 1: The Ten Areas Covered By the Core WS-* Specifications

WS-* Specification Area	Examples
1: Core Service Model	XML, WSDL, SOAP
2: Service Internet	WS-Addressing; WS-MessageDelivery; Reliable Messaging (WSRM); Efficient Messaging (MOTM)
3: Notification	WS-Notification, WS-Eventing
4: Workflow/Transactions	BPEL, WS-Coordination
5: Security	WS-Security, WS-Trust, SAML etc.
6: Service Discovery	UDDI, WS-Discovery
7: System Metadata and State	WSRF, WS-Context WS-MetadataExchange,
8: Management	WSDM, WS-Management, WS-Transfer
9: Policy and Agreements	WS-Policy, WS-Agreement
10: Portals and User Interfaces	WSRP (Remote Portlets)

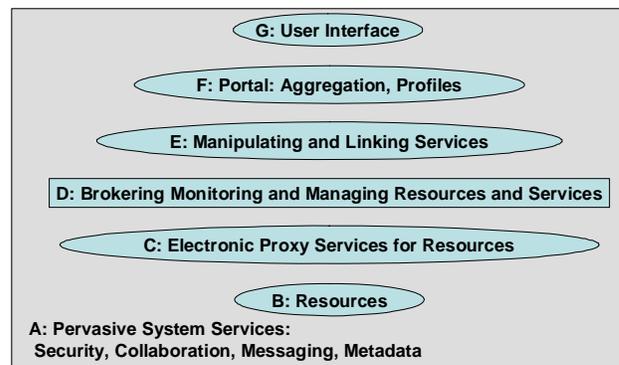


Figure 3. The Grid and Web service functional hierarchy

There are many ways of classifying services, and any classification has grey areas, so it is often possible to move a service between adjacent classifications. For example, security combines handlers for individual messages with sophisticated individual services that support authentication and authorization. Meta-data exists

throughout any system, and in Table 1 (area 7) and Fig. 3(A), we use the term “system metadata,” which is envisaged as the equivalent of Windows registries or UNIX environments. An important class of system metadata specifies aspects of a service and WSRF, the Web Service Resource Framework is important here [8]. Application metadata [9] is equally critical and normally implemented as a queryable database resource. This would be in the language of Fig. 1, set up as a level 4 domain specific service using a level 3 generic Grid database mechanism like OGSA-DAI [10]. As a further illustration of the uncertainties in rigid classifications, Table 1 has separate entries for service discovery and metadata. In fact service discovery is essentially a query to system metadata and included in grey pervasive system services in Fig. 3.

One should note that of the over 60 WS-* specifications, only a fraction have been refined into agreed standards and of these standards only a few have been broadly adopted. The WS-I, or Web Services Interoperability consortium, is the industry group developing consensus on both the accepted standards and how to use them in a set of profiles. Currently < 10% of the WS-* (namely XML, WSDL, SOAP, UDDI and parts of WS-Security) are endorsed by WS-I. There are often competing specifications for a given capability such as the example of two similar specifications WS-ReliableMessaging and WS-Reliability for reliable messaging. We can expect that as experience grows, specifications will be updated, merged and often dropped as a stable endorsed set emerges. Although this process illustrates the immaturity of the field, this is an open, broad, multi-participant activity. We can expect the resultant set of standards to be highly effective and broadly adopted; characteristics of importance to industry and government agencies but that this result can only occur if a broad but necessarily slow process is adopted. Further, the essentials of the resultant architecture are clear – we are typically debating implementation details.

Quality of service and autonomic self-healing are critical Grid characteristics and there is a serious attempt to deal with this in the Grid as the Service Internet (Table 1-area 2) addresses this for messages. Similarly, service management (Table 1-area 8) can be used to build autonomic services. One important aspect of the Web service architecture is that services and messages (not network drivers and packets) are the primitives. Quality of service is thus defined at a higher level than conventional network approaches and we need to understand how to make these approaches blend properly.

Grids assume that the WS-* specifications will mature and be well implemented. Then we need to design and build the services at levels 3 and 4. Correspondingly there are major efforts to design the “important general services” at level 3 and the OGSA (Open Grid Service Architecture) of the Global Grid Forum (GGF) [2] is devoted to this. Note that the community assumes that the first step is to define the open interfaces needed for interoperability for all such common services. The rationale is straightforward as these services cross many application domains (communities of interest) and will involve cooperation between many different developers in business, government and academia.

Table 2: Activities in Global Grid Forum Working Groups

GGF Area	Standards Activities
1: Architecture	High Level Resource/Service Naming (level 2 of Fig. 1), Integrated Grid Architecture
2: Applications	Software Interfaces to Grid, Grid Remote Procedure Call, Checkpointing and Recovery, Interoperability to Job Submittal services, Information Retrieval
3: Computing	Job Submission, Basic Execution Services, Service Level Agreements for Resource use and reservation, Distributed Scheduling
4: Data Access	Database and File Grid access, Grid FTP, Storage Management, Data replication, Binary data, High-level publish/subscribe, Transaction management
5: Infrastructure	Network measurements, IPv6 and high performance networking, Data transport
6: Management	Resource/Service configuration, deployment and lifetime, Usage records and access, Grid economy model
7: Security	Authorization, P2P and Firewall Issues, Trusted Computing

The loose coupling of web services requires no agreement on service implementation but it does require agreed interfaces so that the SOAP messaging can communicate between services from different sources. Thus we expect that in levels 3 and 4 one needs major attention to standards for services and data. The standards in level 3 must be broadly endorsed while those in level 4 of Fig. 1 must be endorsed by the applicable community of interest. OGSA currently divides the services in level 3 into categories, which are perhaps easiest to classify in

terms of the GGF work areas given in first column of Table 2 with some examples of their work in column 2. The above table is illustrative of the international activity setting level 3 standards. There is much additional work within individual projects and organizations and within other standards organizations. A good example of the latter is the work of the Open Geospatial Consortium on Geographical Information System (GIS) services [11, 12]. This also illustrates that different communities might classify services into different levels. Sensor-based applications could view GIS as a universal level 3 service and job submittal (a major focus of GGF) as rather specialized and so level 4. Clearly the Global Grid Forum views job processing as central and so far has not looked at GIS in detail.

Table 3: Core Global information Grid Net Centric Services

Label	Service or Feature	Examples
NCES 1	Enterprise Services Management	Life Cycle Management
NCES 1	Security; Information Assurance (IA)	Confidentiality, Integrity, Availability, Reliability
NCES 3	Messaging	Publish-Subscribe important
NCES 4	Discovery	Data and services
NCES 5	Mediation	Agents, Brokering, Transformation, Aggregation
NCES 6	Collaboration	Synchronous and Asynchronous
NCES 7	User assistance	Optimize GiG experience
NCES 8	Storage	Retention, Organization and Disposition of all forms of data
NCES 9	Application	Provisioning, Operations and Maintenance
ECS	Environmental Control Services	Policy

The US Department of Defense has developed the Net Centric Environment (NCE) as a future architecture for defense information systems with their Global Information Grid (GiG) as the infrastructure on which the NCE is to be deployed. There are nine core services defined for the NCE given in table 3 together with an additional policy layer. Note NCE recognizes the importance (and inevitability) of service oriented architectures but do not define a clear relationship of their architectures to Grid or Web services. Also we note that DoD explicitly identifies collaboration as a core service but this does not directly appear in the earlier tables.

If we synthesize the work of the Grid and NCE communities, we can define in Table 4, a set of 18

features and services that form levels 1, 2 and 3 of Fig. 1 and on which domain specific services (level 4) can be built.

Table 4: 18 Categories of Core Features and Services

Service or Feature	WS-*	GS-*	NCES	Comments
A: Broad Principles				
FS1: Use SOA: Service Oriented Architecture	WS1			Core Service Model, Build Grids on Web Services. Industry best practice
FS2: Grid of Grids	Strategy for legacy subsystems and modular architecture			
B: Core Services				
FS3: Service Internet, Messaging	WS2		NC3	Streams/Sensors
FS4: Notification	WS3		NC3	JMS, MQSeries
FS5: Workflow	WS4		NC5	Grid Programming
FS6 : Security	WS5	GS7	NC2	Grid-Shib, Permis Liberty Alliance ...
FS7: Discovery	WS6		NC4	
FS8: System Metadata & State	WS7			Globus MDS Semantic Grid
FS9: Management	WS8	GS6	NC1	CIM
FS10: Policy	WS9		ECS	
FS11: Portals and User assistance	WS10		NC7	Portlets JSR168, NCES Capability Interfaces
FS12: Computing		GS3		
FS13: Data and Storage		GS4	NC8	NCOW Data Strategy
FS14: Information		GS4		JB1 for DoD
FS15: Applications and User Services		GS2	NC9	Standalone Services Proxies for jobs
FS16: Resources and Infrastructure		GS5		Ad-hoc networks
FS17: Collaboration and Virtual Organizations		GS7	NC6	XGSP, Shared Web Service ports
FS18: Scheduling and matching of Services and Resources		GS3		

Note in Table 4 that WS_n ($n=1..10$) refer to rows of Table 1; GS_n ($n=2..7$) refer to rows of Table 2. NC_n and ECS refer to rows of Table 3. We start Table 4 with two important principles with the (Web) service backbone (FS1) corresponding to the row WS-1 of Table 1. We have discussed the Grid of Grids concept (FS2) at length in [1] and suggest this is one approach to both legacy systems and modular architectures. One cannot realistically build a single monolithic Grid with all services compliant to the same set of standards. We will always need to federate systems together with different internals. One structures each system as an individual Grid and using mediation technology (with at its simplest

running through parameter spaces. We also see in Fig. 5 sensor and instrument services that as in Fig. 4 are constructed so that their output streams are similar to those of filters and simulations and would be viewed as level 4 services or if you prefer a part of FS13. Similarly visualization is very important for many Grids and this would be a level 4 service building on FS11 portals. In many applications we have found a GIS Grid very important [12] as a set of level 4 services. The collaboration and community services discussed in the following sections can either be called part of FS17 or define a subset of themselves as the core FS17 on which to build a suite of level 4 services. At this stage the distinction between the different levels in Fig. 1 is not so important. More interesting is the types of services needed and how they interact together; we join tightly coupled services together into (sub)Grids and compose these for the full Grid with comparatively looser coupling.

3. COLLABORATION SYSTEMS

Collaboration is all about “togetherness” and this involves some form of sharing but there are many variants of this as we will discuss below. Of course communities just describe those that are “together” or who are collaborating. The traditional Grid application involves sharing of computers (such as large supercomputers on the TeraGrid), networks and data repositories. Here there is some togetherness as maybe we need to manage multiple sharing files or computer allocations. However a lot of the problem is separateness as we need to take resources and divide them up to minimize interference between different users. We note that this class of Grid applications extends shared data and file repositories seen in both pervasive technologies like WebDAV and customized systems such as KnowledgeKinetics [14]. The classic Internet exemplifies another form of collaboration where information is posted by one individual or organization and accessed by everybody else or perhaps those selected through some authorization scheme. Simple Internet pages are very important in creating communities with for example the Olympic web pages supporting both the host country and particular groups of sports fans. The page listing my research group’s publications supports the Community Grids Laboratory community; more importantly the exciting field of Bioinformatics is some ways enabled by the many

internet resources like the Protein Data Bank and PubMed. Note in this type of togetherness there are not direct interactions between the members of a community but rather individual to resource and resource (web page and database) to member interactions. An important variant of the shared Internet resource is seen in the many file sharing systems where the original Napster, Gnutella and BitTorrent are representative technologies. It is worth noting that this class of application is a far larger contribution to Internet traffic than all the Grid applications.

The traditional synchronous collaboration involves peer-to-peer or server-client (resource-community member) interactions that both are sharing resources (as in the Internet classic) but also their updates. Here we can remember Metcalfe’s law that the value of a network is proportional to number of links so that with N members of a community, a resource such as a web page has a value proportional to N . On the other if we link M members in a typical collaborative session, then the value is proportional to M^2 . Of course values of M are typically smaller than the N appropriate for Internet pages although typical Grids have modest size N . Nevertheless because asynchronous collaboration can be done at any time, it essentially always has a much larger size than the synchronous case. The synchronous type of collaboration includes shared applications with a variety of techniques including shared display and shared event. There are also specialized tools such as audio-video conferencing, white boards, text chat and instant messengers. In a Grid model, shared applications become shared services while the specialized tools become particular services in a Collaboration Grid [15]. Of course the shared application case is a little problematic in important cases where the applications as illustrated by PowerPoint are not services. The problem of making an application collaborative using shared events is equivalent to packaging it as a Web Service with ports to define (user) input and events defining the state change in event. We have recently examined this for SVG [16], PowerPoint[17], OpenOffice [18] and an interactive language IDL [19]. Skype, the audio and now video offshoot of Kazaa, is clearly an important collaborative application and emphasizes the value of P2P technologies in this problem class. The many Internet games available often support a synchronous collaboration shared event model and multi-player games are an interesting Collaboration Grid.

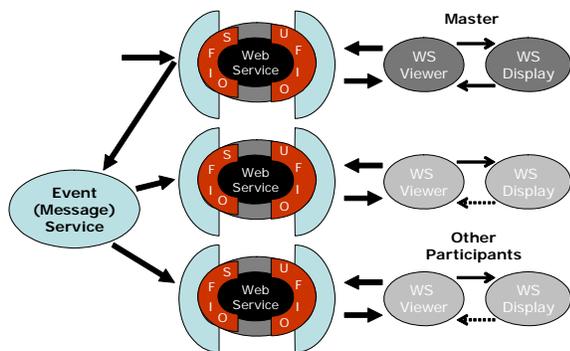


Figure 6. Shared Input Port (Replicated WS) Collaboration. UFIO and SFIO are User Facing and Service Facing Input/Output Ports

There are many interesting community tools starting with email, which is still for instance the dominant form of electronic collaboration in most of my activities. Many community tools are variants of the shared Internet resource but allow ways for multiple people to contribute to that resource. These include venerable bulletin boards, list-serves or forums and more recently Wikis where multiple people can edit a set of web pages. In these and other cases one can either allow a free for all or establish an authorization system to control those who can change or view the site. Blogs are an important variant of a Wiki where typically the basic material is produced by an individual but comments can be made by others. Sites such as <http://www.hotornot.com> support shared photographs and their annotation. Annotation has proved to be extremely popular in the sites supporting generalized bookmarks with excellent reviews in Refs. [20] and [21]. <http://del.icio.us/> supports annotation and sharing of bookmarks while <http://www.connotea.org>, <http://www.bibsonomy.org/> and <http://www.CiteULike.org> add features relevant for document collections with the latter two supporting the familiar Bibtext constructs. A striking feature of all these sites is their simplicity – the simple keyword annotation chosen by these sites (and their customers) is a far cry from the sophistication of RDF and certainly OWL in the Semantic Web. Such community tools are customized for different domains and applications as for example LinkedIn <http://www.linkedin.com>, which uses metadata to form people networks (i.e. communities). Tools such as Citeseer and Google Scholar identify citations in articles and so can form communities (and hence collaborations) by identifying people with common interests. More generally domain-specific web crawlers can link documents by common application meta-data and so link both the authors and subjects of the web resources. We are exploring this in Chemical Informatics <http://www.chembioGrid.org> where the labels (names) of chemical compounds allow this linkage. Applications like

e-learning or e-business need a mix of above basic collaboration models. For example e-learning might use a mix of synchronous delivery (using in simple cases shared PowerPoint and audio-video conferencing) and an asynchronous learning management system.

4. COLLABORATION GRIDS

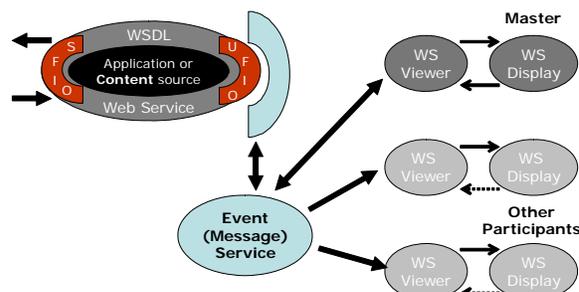


Figure 7. Shared Output Port (Single WS) Collaboration that is possible at any point on visualization pipeline

In this section, we discuss Grid architectures supporting collaboration and communities. First we recall the simple idea [16] [22] allowing the sharing of Web services. This is shown in Figs. 6 and 7 for the two basic cases of replicated or single services. We exploit the definition of a Web Service which implies that it is defined by its input messages and exhibits the impact of the input messages on its output ports. In Fig. 6 – termed MMMV (Multiple Model Multiple View) by Qiu [16] – we instantiate multiple identical services. We can ensure that they track each other as needed say by multiple copies of a PowerPoint presentation in e-learning, by ensuring they get the same messages. These messages are always explicit in a (Web) service architecture, and we just need to replicate the input messages. This multicast is enabled by NaradaBrokering [23] in our applications. Jabber technology [24] is also a common choice for this type of capability. In Fig. 7, we show the shared output port or Single Model Multiple View SMMV [16] collaboration model. Note that shared display case falls into this category with the messages corresponding to the final stage of the visualization pipeline when they define pixel changes in the framebuffer. This approach to collaboration applications can straightforwardly make any service collaborative and corresponds to a framework that is supported by the message delivery system [23] in FS3 and a “session manager” which is a service in FS17 in the terminology of Table 4. One can view a session manager as “just” a domain (here collaboration) specific metadata handler which records the users and services (applications) involved in a particular collaborative session. In accordance with the information FS14 model [13], such a metadata system needs a custom language and should probably store the metadata in a database

“on the edge”. However we don’t see any difficulty in implementing any peer-to-peer community system with service architectures and think it is important to support peer-to-peer messaging, transport (BitTorrent), security and discovery models for Grids. The gateways we discussed above for GlobalMMCS and community tools in the SSG, exemplify how one can build service interfaces to non Grid systems. Further the use of NaradaBrokering in both Grid and JXTA [39] illustrates that one can capture differences between Grid and peer-to-peer systems in different implementation of the core (messaging) services. Further Skype and GlobalMMCS offer similar capabilities and the huge success of the former, suggests that peer-to-peer implementations can be very competitive with the more traditional server-based solutions for large scale robust applications.

5. CONCLUSIONS

In this paper we have shown how one can collect all the desirable community and collaboration tools together and wrap them as services to form a Collaboration Grid. This can be customized to support different communities and deliver a next generation of portals and laboratories. One can choose between peer-to-peer and server based implementations of the system.

REFERENCES

[1] Typical Grid references are *The Grid 2: Blueprint for a new Computing Infrastructure*, edited by Ian Foster and Carl Kesselman, Morgan Kaufmann 2004 and *Grid Computing: Making the Global Infrastructure a Reality* edited by F. Berman, G. Fox and A. Hey, Wiley, March 2003.

[2] Global Grid Forum <http://www.Gridforum.org>

[3] Geoffrey Fox, Alex Ho, Shrideep Pallickara, Marlon Pierce, Wenjun Wu *Grids for the GiG and Real Time Simulations* <http://Grids.ucs.indiana.edu/ptliupages/publications/gig/DSRTOct05.pdf> Proceedings of Ninth IEEE International Symposium DS-RT2005 on Distributed Simulation and Real Time Applications, pages 129-138 IEEE Computer Society <http://doi.ieeecomputersociety.org/10.1109/DISTRA.2005.23>, Montreal October 10-12 2005

[4] Geoffrey Fox, Alex Ho, Marlon Pierce Collection of Grid Resources for DoD Applications, Internal Reports June 2005, <http://Grids.ucs.indiana.edu/ptliupages/publications/gig/>.

[5] Ken Birman, Robert Hillman, Stefan Pleisch, *Building network-centric military applications over service oriented architectures* SPIE Conference DEFENSE TRANSFORMATION AND NETWORK-CENTRIC SYSTEMS Orlando Florida 31 March 2005. http://www.cs.cornell.edu/projects/quicksilver/public_pdfs/GIGonWS_final.pdf

[6] Booth, D. et al.. *Web Service Architecture* W3C Working Group Note, 11 February 2004. <http://www.w3c.org/TR/ws-arch>

[7] Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Tony Storey, Donald F. Ferguson, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*, Prentice Hall March 22, 2005, ISBN: 0-13-148874-0.

[8] Globus Project <http://www.globus.org>.

[9] Semantic Grid <http://www.semanticGrid.org>

[10] OGSA-DAI Grid and Web database interface <http://www.ogsa-dai.org/>.

[11] Open Geospatial Consortium <http://www.opengeospatial.org/>

[12] GIS Grid Services <http://www.crisisGrid.org>

[13] Geoffrey Fox *Services and the Semantic Grid* Panel discussion at SKG 2005 International Conference on Semantics, Knowledge and Grid Beijing Friendship Hotel November 28 2005. <http://Grids.ucs.indiana.edu/ptliupages/presentations/skg05panelnov28-05.ppt>

[14] KnowledgeKinetics, Ball Aerospace enterprise collaboration environment http://www.ball.com/aerospace/k2_home.html

[15] GlobalMMCS Service oriented Collaboration Environment from Community Grids Laboratory <http://www.globalmmcs.org>.

[16] Xiaohong Qiu *Message-based MVC Architecture for Distributed and Desktop Applications* Syracuse University PhD March 2 2005 <http://Grids.ucs.indiana.edu/ptliupages/publications/qiuPhDthesis.pdf>

[17] Minjun Wang, Geoffrey Fox and Shrideep Pallickara *Demonstrations of Collaborative Web Services and Peer-to-Peer Grids* Journal Of Digital Information Management. Volume 2, Issue 2. June 2004. Special Issue of selected papers from the IEEE ITCC 2004 Track on Modern Grid and Web Systems. http://Grids.ucs.indiana.edu/ptliupages/publications/P2PGrids_JDIM_PDF.pdf

[18] Minjun Wang and Geoffrey Fox *Design of a Collaborative System for Open Office*, Proceedings of IASTED KSCE Conference Virgin Islands November 2004 <http://Grids.ucs.indiana.edu/ptliupages/publications/OpenOfficeCollaborativeSystemFINAL.pdf>

[19] Minjun Wang, Geoffrey Fox and Marlon Pierce *Grid-based Collaboration in Interactive Data Language*

- Applications* Proceedings of International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I pp. 335-341, April 4-6, 2005, Las Vegas http://Grids.ucs.indiana.edu/ptliupages/publications/GridCollabIDL_ITCC2005.pdf
- [20] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott *Social Bookmarking Tools (I) A General Review*, D-Lib Magazine April 2005 Volume 11 Number 4 <http://www.dlib.org/dlib/april05/hammond/04hammond.html>
- [21] Ben Lund, Tony Hammond, Martin Flack and Timo Hannay *Social Bookmarking Tools (II) A Case Study - Connotea*, D-Lib Magazine April 2005 Volume 11 Number 4 <http://www.dlib.org/dlib/april05/lund/04lund.html>
- [22] 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 *Grid Computing: Making the Global Infrastructure a Reality* edited by F. Berman, G. Fox and A. Hey, Wiley, March 2003.
- [23] NaradaBrokering Grid Messaging <http://www.naradabrokering.org>
- [24] Jabber Software Foundation <http://www.jabber.org/>
- [25] Wenjun Wu, Geoffrey Fox, Hasan Bulut, Ahmet Uyar, Tao Huang *Service Oriented Architecture for VoIP conferencing* to be published 2006 in Special Issue on Voice over IP - Theory and Practice of the International Journal of Communication Systems <http://Grids.ucs.indiana.edu/ptliupages/publications/soa-voip-05.doc>
- [26] Ahmet Uyar *Scalable Service Oriented Architecture for Audio/Video Conferencing* Syracuse University PhD March 23 2005 <http://Grids.ucs.indiana.edu/ptliupages/publications/UyarThesisFinal.PDF>
- [27] GGF14 Information Retrieval Grid Session <http://www.ggf.org/GGF14/GGF14IR.html>
- [28] Citeseer citation analysis system <http://citeseer.ist.psu.edu/>
- [29] Del.icio.us annotation system <http://del.icio.us/>
- [30] Connotea annotation system <http://www.connotea.org/>
- [31] CiteUlike annotation system <http://www.citeulike.org/>
- [32] GGF9 workshop on Peer-to-Peer and Grids: Synergies and Opportunities <http://csag.ucsd.edu/P2P-Grid/>
- [33] Ian Taylor, The Triana Project <http://www.trianacode.org/index.html>
- [34] Kai Hwang USC Peer-to-Peer Grids, <http://Gridsec.usc.edu/Hwang.html>
- [35] Shanshan Song, Kai Hwang, Runfang Zhou, Yu-Kwong Kwok, *Trusted P2P Transactions with Fuzzy Reputation Aggregation*. <http://csdl.computer.org/dl/mags/ic/2005/06/w6024.pdf>
- [36] Ian Foster, Adriana Iamnitchi, *On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing*, http://people.cs.uchicago.edu/~anda/papers/foster_Grid_vs_p2p.pdf
- [37] Adriana Iamnitchi, Matei Ripeanu and Ian Foster, *Small-World File-Sharing Communities*, INFOCOM 2004, March 2004, Hong Kong. <http://arxiv.org/abs/cs.DC/0307036>
- [38] M. Cai, A. Chervenak, M. Frank, *A Peer-to-Peer Replica Location Service Based on A Distributed Hash Table*. Proceedings of the SC2004 Conference (SC2004), November 2004 <http://www.globus.org/alliance/publications/papers/sc2004v15.pdf>
- [39] Geoffrey Fox, Shrideep Pallickara, Xi Rao *Towards enabling peer-to-peer Grids* Concurrency and Computation: Practice & Experience archive Volume 17, Issue 7-8 (June 2005) Pages: 1109 - 1131. 2002 ACM Java Grande-ISCOPE Conference Part II <http://Grids.ucs.indiana.edu/ptliupages/publications/CCPE-P2PGrids.pdf>
- [40] Domenico Talia, Paolo Trunfio, *Toward a Synergy Between P2P and Grids*, <http://csdl.computer.org/dl/mags/ic/2003/04/w4096.pdf>
- [41] Robson Santos, Francisco Brasileiro, Alisson Andrade, Nazareno Andrade, Walfredo Cirne, *Accurate Autonomous Accounting in Peer-to-Peer Grids*, <http://middleware05.objectweb.org/WSProceedings/MGC05/a10-Santos.pdf>