A Survey of the Role and Use of Web Services and Service Oriented Architectures in Scientific/Technical Grids

Geoffrey Fox[1,2,3], *Dennis Gannon*[2]

[1]Community Grids Laboratory, Indiana University
501 N Morton Suite 224, Bloomington IN 47404

[2]Computer Science Department and School of Informatics, Indiana University

[3]Physics Department, Indiana University

Executive Summary

Web Services and Service Oriented Architectures are now a fundamental part of the design of current Grids. In this report we look at the current state-of-the-art based on an analysis of the major Grid efforts in the US and Europe. We distinguish two classes of Grids. The first class consists of “general purpose” Grids that provide computing and data resources to a broad class of application communities: EGEE, TeraGrid, Open Science Grid and Naregi. The second category of Grids are those devoted to a specific scientific or technical application field, such as geosciences, chemical informatics, earthquake science, astronomy and atmospheric science. In the first category, the use of services is based on providing the foundational elements of security, data management, remote job execution and information services. While in the second category we find more specialized services including application services, user-level metadata services, data discovery services and specialized workflow tools. We have categorized the services used in these Grids against services hierarchies defined by the Department of Defense, OGF, W3C and Oasis.

In general there is rough consensus on the very basic web service building blocks, the WS-I profile of services and standards, WSDL, SOAP, WS-Addressing. But beyond these, there is spotty use of the other WS-* standards. This is, in part, due to the fact that many of these additional standards are new, and robust and efficient implementations are only now becoming available. Globus GT4 and large projects like gLITE are widely used to provide the additional foundational elements such as security and data and job management, and the domain-specific Grids are building the application specific services they need on top of the foundational tools as opposed to on top of a suite of services compatible with standards. While there is substantial repetition of design and functionality from one domain-specific Grid service stack to the next, there has been very little movement towards standardization with common building blocks as considered in OGF. This seems to be partly due to the fact that these domain Grids have not had a need to interoperate and partly due to lack of time to get experience and consensus. As science becomes more interdisciplinary and grids become more mature, this may change.

However, there are areas where a substantial amount of technology sharing is taking place. Beyond the core service areas and packages mentioned above, portal technology and workflow tools are widely shared. Because of the growing significance of workflow tools in Grids, we have included a section of this report on that area. Workflow tools depend on a number of support services to work in Grid environments. Again, although there is no standard workflow tool, there are many common issues that must be addressed to make workflow systems actually successful. Security, fault tolerance, rich application metadata, provenance are among these. While it may not be necessary or possible to agree on standard services, there are a number of places where standard metadata will greatly facility interoperability.

In the area of services, what is most important to working scientists is trusted, running code. Standards emerge from rough consensus on code or tool, or when interoperability and portability is required and that is not possible without agreement.

Recommendations

The current progress on building and deploying substantial Grid is strong. There are almost as many Grid software stacks as there are Grids. The most commonly used pieces of infrastructure consists of the WS-*

core services, Globus GT*, Condor, SRB and gLite. A great deal of additional infrastructure has been built on top of these core tools. Much of this additional infrastructure involves wrapping legacy systems. The largest problems these Grids will face involves sustainability and interoperability. For many of the large government-funded Grids, when the funding for development runs dry, maintaining and extending the infrastructure becomes extremely difficult if they are based on stove-pipe custom designs. When the need arises for these Grids to interoperate so that services can be shared, the custom nature of each will hinder progress in this area. Standards play a role as long as they are imbedded in trusted, shared software frameworks which can be supported commercially or through a large open-source community. If we consider where we will be in five years, there are several areas where some effort can produce some significant results.

1. We recommend continuation of work in the WS-* core services with any needed steps to encourage progress typified by the recent standards mergers. As soon as this is resolved, debate about WSRF can end. The popular REST approach should be expected to continue and appropriate recognition of this is recommended. Also, it is important for the technical community to understand the implication of WSDL 2 and its support by Axis2 for Java.
2. The WS-* services have modest adoption at present but we can expect this to improve naturally as core software matures and standards convergence occurs. For some time we will need multiple stacks but this number will decrease and interoperability improve in the future. We recommend work on interoperability of core stacks to ease transition and that this work include REST and WS-* protocols.
3. For the higher level standards we recommend that more attention be given to the interoperability between large systems that may or may not be built in terms of Grid services and may or may not be standards compliant. This is illustrated by Condor, SRB, GT4 or the many separate Astronomy collections for e-Science; by BPEL, Pegasus, Taverna and Triana in workflow; digital libraries represent important data grid-like systems whose grid standards integration should be clarified; in the military the concept of "systems of systems" is used to describe system (sub-grid) interoperability and integration. We recommend that one needs develop bottom up computing and data standards but as well the higher level managed computing (workload) and managed data (information-knowledge-wisdom) interoperability. The implications for the OGSA road-map should be considered. We now go into more detail below.
4. Data Management Services represent a clear opportunity for developing common solutions. Many pieces of core technology exist. These include SRB, OGSA-DAI, VOSpace and the Globus Data Replication Service (DRS). Other technology exists on the web to support wide area data management, such as BitTorrent and many corporations are providing data hosting services. We recommend identification of best practice and needed specifications to support Data Virtualization, which is an important core capability. What is needed is a service that allows a VO to manage all their data objects in a way that frees the users and client services from the details of storage management. This should support data replication and high-speed data transfer when a user wishes to resolve a name into a concrete data object.
5. Data Management should be extended to Knowledge Management and we recommend studies of best practice and needed standards that will integrate metadata and support federation at all points on the data-information-knowledge-wisdom pipeline. The

VOSpace literature discusses differences between storage (resource view) and knowledge (user view) management.

6. Building a comprehensive, Application Metadata specification is an area that is critical to making services useable by many workflow systems. We recommend the development of a standard schema to describe the semantics of application services. This will enable the next generation of workflow tools to reason about how individual services are invoked.
7. The current UDDI specification is a key part of WS-I but essentially all grids have found it inadequate (see Sec. 1.3) and there are many enhanced versions of it. We recommend an activity to either develop a common Grid-enhanced UDDI or work with the UDDI to community to address its limitations.
8. We recommend continued study of data and workflow provenance. While this still may be a research topic, it may soon be possible to provide a reusable standard schema for data provenance for technical computing.
9. We recommend continued attention to the user view of grids including workflow and portals (gateways) which provide important interoperability interfaces.
10. The OGSA roadmap correctly identifies the importance of resource and self management and we recommend further research in this area which seems critical for providing robust scalable grids.

1. Introduction to Service Analysis

We used a refinement of analysis originally developed to understand Dod's Net-Centric architecture (section 2.1) in [DoD1] with this also summarized in [DOD2]. Here we divided service architecture components into four as shown in Figure 1. Layer 1 is the container and hosting environment where in eScience, Apache Axis or .NET are typically used. There is general agreement at the functionality here although some of the technology is still not very mature. The next layer is that of the WS-* divided into ten categories in table 1. These are the core service standards (specifications) and are set dominantly in W3C and OASIS. More details on the services in each category can be found in catalog [DoD3].

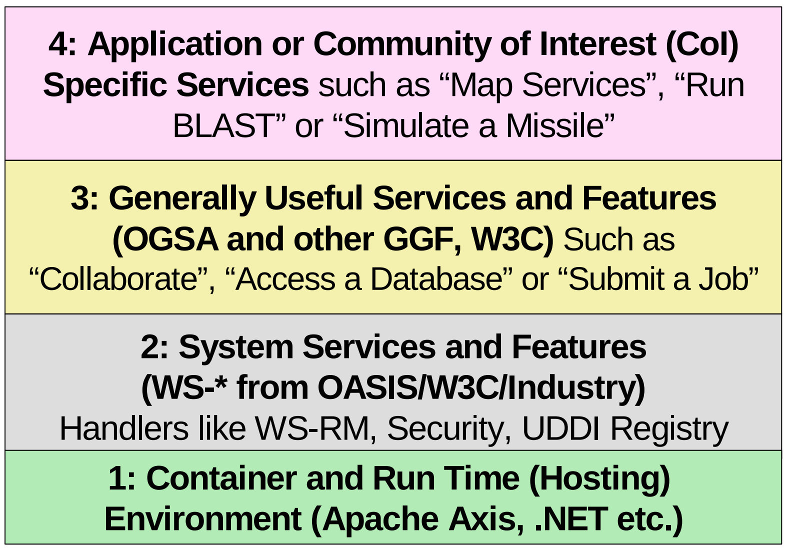


Figure 1. The Grid and Web service Institutional Hierarchy

Layer 3 contains what is termed "generally useful services" which are higher level than the core services but used in multiple Grids. Examples include "Submit a Job":, "Access a Sensor" or "Collaborate". Here the Open Grid Forum standards activities are very important and so in table 2, we summarize GGF (OGF) work using the functional areas defined by OGF. Alternatively, we could have used the organization described by the OGSA architecture. These seven OGF areas are referred to as GS-* (*=1 to 7) below. Table 3 gives another cut on the division of a distributed system into service areas with a list of the 10 core areas (Enterprise Services) defined by DoD for their Global Information Grid. Table 4 is the main result in this section as it lists eighteen "features and services" that encompass current Grid applications, Net-Centric computing and the current work of W3C, OGF, OASIS and DMTF. As discussed in DoD analysis cited above, there are reasonable agreement between "GGF style" (OGSA) and DoD Net-Centric architectures. DoD does not put computing related services in their core but does highlight synchronous and asynchronous collaboration. Collaboration appears in OGSA style Grids as virtual organizations [PicklesItaly] [DoD2] but the services of interest to DoD have not been highlighted in OGF. AFRL (Air Force Research Laboratory) has defined a Collaborative Enterprise Environment CEE [CEE00] which has been implemented by Ball Aerospace [KK]. The CTS series of conferences [CTSxx] covers collaboration issues of interest to DoD. Typical DoD application is "Command and Control" (called Emergency Response in civilian case) and here collaboration is between commanders/first responders to make decisions in real-time. Scheduling is discussed in GGF for computing but not for networks and services as needed by DoD. DoD also endorses the "Grids of Grids" architecture (termed by them systems of systems) as it allows modular development and the wrapping of "legacy" systems as grids. Table 4 includes all the broad service categories needed by the grids discussed here and all the analyses in section 2 reference the categories in this table. We discuss later confusion in the use data and information and we explicitly separate "raw and stored" data from "interpreted or managed" data for which we (and DoD) prefer the term information. However in conventional data grid terminology, information is used to describe monitoring. We distinguish these different concepts in our analysis. Finally note that layer 4 in figure 1 corresponds to the domain specific (community of interest) services that are not discussed much here as it is expected that these would be developed separately by each field. The "general" services of table 4 include capabilities in FS13-15 that are important in building domain specific services and further all the core services FS1-11 (especially workflow) will be important here. However the responsibility for building these services of layer 4 in fig. 1 lies with domain and not with the "central IT institutions".

Section 2 analyses 68 different grids in terms of services and section 3 provides an analysis of workflow tools. Section 4 gives some conclusions.

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, Messaging, Routing, addressing	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)

Table 2: Activities in Global Grid Forum (OGF) 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

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 user 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

NCES refers to DoD Net-Centric Enterprise Services

Table 4: Summary of 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 (Mainly Service Infrastructure and W3C/OASIS focus)				
FS3: Service Internet, Messaging	WS2		NCES3	Core transport service to service, REST implies HTTP not SOAP
FS4: Notification	WS3		NCES3	JMS, MQSeries, WS-Eventing,Notification
FS5 Workflow	WS4		NCES5	Grid Programming
FS6 : Security	WS5	GS7	NCES2	GSA, Grid-Shib, Permis Liberty Alliance ...
FS7: Discovery	WS6		NCES4	UDDI, GT4 index and many custom services
FS8: System Metadata & State	WS7			Globus MCS, WSRF, WS-MetadataExchange, Semantic Grid, REST has stateless paradigm
FS9: Management	WS8	GS6	NCES1	CIM, WSDM, WS-Management
FS10: Policy	WS9		ECS	WS-Policy
FS11: Portals and User assistance	WS10		NCES7	Portlets JSR168, NCES Capability Interfaces
C: Generally useful Services (Mainly Higher level and OGF focus)				
FS12A: Core Computing		GS3		Job Submittal and Scheduling
FS12B: Managed Computing or “Workload management”		GS3		Including services such as organize “parameter search” or related jobs such as those analyzing a group of LHC events
FS13A: Data and Storage		GS4	NCES8	Parts of NCOW “Data” Strategy. Here we see files and databases. The metadata used often to adorn “raw data” can be considered as higher level FS14B RLS, OGSA-DAI and SRM
FS13B: Streams and Sources/Sensors				Sources could include monitors whose content classified in FS14A SensorML from OGC
FS13C: Data Transport		WS2 WS3 GS4	NCES3 NCES8	GridFTP or WS Interface to non SOAP transport, Globus RTF
FS14A: Information as Monitoring		GS4		In a “data grid” world, the results of a monitoring source such as a job status update is called “information” as in GMA Grid Monitoring Architecture
FS14B: Information, Knowledge, Wisdom part of D(ata)IKW **Can be considered as “Managed Data” combining data and metadata**		GS4	NCES8	VOSpace for IVOA, JBI for DoD, WFS for OGC and parts of NCOW “Data” Strategy Federation at this layer major research area If we take a traditional DIKW hierarchy – Data, Information, Knowledge, Wisdom – then we use FS14B to describe higher level IKW

			services and FS13A for lowest level SRB
FS15: Applications and User Services	GS2	NCES9	Standalone Services Proxies for jobs
FS16: Resources and Infrastructure	GS5		Ad-hoc networks
FS17: Collaboration and Community services	GS7	NCES6	XGSP, Shared Web Service ports Advanced aspects of Virtual Organizations
FS18: Scheduling and matching of Services and Resources	GS3		This describes scheduling services – not the scheduling of computing jobs included in FS12 which is a major part of Grid activity

NCES refers to DoD Net-Centric Enterprise Services in Table 3
WS-* refers to Web services standards in Table 1
GS-* refers to OGF activities in Table 2

1.2 Notes on Data and Information Architecture

This elaborates on the topics labeled FS13 and FS14 in table 4. The data area is confused due to its diversity compounded by inconsistent use of terminology. We had noted that what is often called a data grid (as in EDG European Data Grid or gLite discussed here) was rather different from the database oriented Grids seen in IVOA (discussed here) and bioinformatics grids like MyGrid. We attempted to re-name them Compute-File and Information grids respectively in the Gap Analysis with Walker [GapAnalysis] but this terminology was not popular. In figure 2, we depict a general scenario. In all fields there is some transition often called DIKW through data, information, knowledge and wisdom. Always there are many processing steps and one hopes that the value of the bits produced increases so that one progresses up the D-I-K-W chain. Of course there is no rigorous division between the DIKW levels and one group could label as knowledge what another group labels information. Further one might use decision (support) as the name of the final stage rather than wisdom. There is of course similar

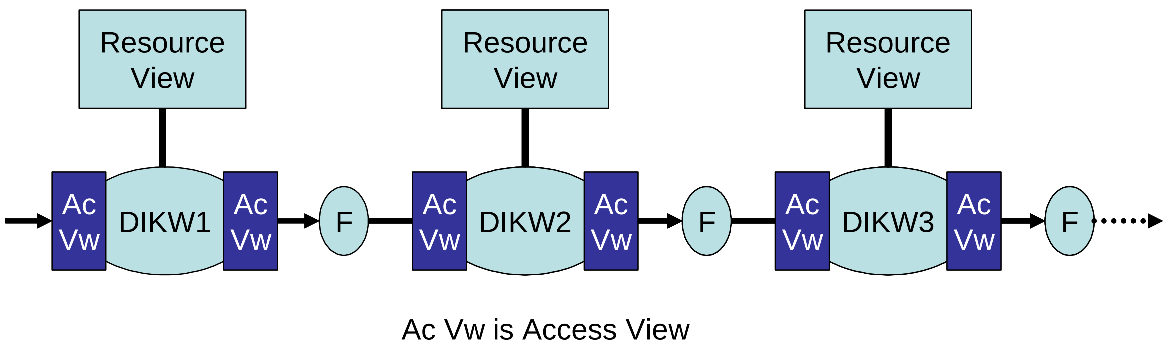


Figure 2: Data-Information-Knowledge-Wisdom Pipeline. DIKWi represent different forms of DIKW with different terminology in different fields. Each DIKWi has a resource view describing its physical instantiation (different distributed media with file, database, memory, stream etc.) and an access view describing its query (input or output) model (dir or ls, SQL, XPATH, Custom etc.). The different forms DIKWi are linked by filtering steps F. This could be a simple format translation; a complex calculation as in the running of an LHC event processing code; a proprietary analysis as in a Search engines processing of harvested web pages; an addition of a metadata catalog to a collection of files.

ambiguities between data and metadata; one person's metadata is another person's data. At each step of transformation chain shown in fig. 2, one can distinguish the user (access) and resource view. The user view focuses on the meaning of the DIKW while resource view on the mechanisms and physical devices used to store. The resource view includes issues such as storage device, file-caching, distribution and whether a database, structured or unstructured file or files are used. Other resource mechanisms are real-time streams, notification events, sensor outputs, and in-memory. The access view could vary from pure raw data, data plus a metadata catalog. SQL, XPATH to application specific syntax (VOQL in astronomy, WFS in GIS). Note both the resource view and access view can be exposed as Grid services but in some cases you have no choice as existing systems only expose the access view as in GIS systems that have database backends (resource views) but the exposed interoperability interface is the access view WFS.

SRB, SRM and OGSA-DAI are important relevant grid technologies. SRM focuses on virtualizing a file-oriented resource view to hide the details of different storage devices; OGSA-DAI virtualizes many different database paradigms while supporting directly transformations to different access views (filters in fig. 2). SRB addresses the access view but with a focus on collections of files. GLite has a data management system that includes SRM functionality, caching and a metadata catalog to support a file-oriented access view. Note that in some parlance, GLite "manages data" to produce next step in DIKW pipeline but it is confusing to call this step information as this term is typically used in GLite and related data grids to refer to resource and job related (meta)data.

DIKW federation (integration of multiple services with the same access view) can be implemented at both access and resource view and at different points on pipeline. There is clearly substantial work in the database community on this point and OGSA-DAI implements as DQL for the Grid. However this resource view solution cannot be applied in cases where only the access view is available for federation. Here SRB would approach federation at the access level (a metadata catalog and typically a collection of files) by integrating the metadata catalogs of the federation components.

1.3 UDDI Universal Description, Discovery and Integration

All web-service based grids need some sort of service discovery and several have evaluated the WS-I approved UDDI standard. However essentially all have found UDDI inadequate and below we list the reasons for this. Briefly UDDI is not metadata-oriented (see points 1,2 below). It may contain stale information about Web Service entries (see point 4). It does not support session metadata (see point 3). It has a poor service description and inquiry matchmaking process (see point 5). The UDDI does not address to domain-specific needs of various domains such as GIS (see point 6). It is a centralized registry (see point 7) which like point 4 is a comment on current implementations rather than the interface specification. Of course some of the e-Science limitations reflect the e-Business focus of UDDI but many of the suggested improvements from the Grid community should be beneficial in all applications of UDDI.

1) Out-of-box UDDI [UDDI] is not metadata oriented. It does not support the ability to publish and search prescriptive metadata of services. The prescriptive metadata is the domain-specific information about the functionality of a service.
2) In the same line with previous point (it is not metadata-oriented), the UDDI Specs does not take into account descriptive metadata, i.e. quality of service attributes, into it's discovery process. The prescriptive metadata provided by a service may be fitted with client's request, however, this does not necessarily guarantee whether the service is sufficient for the desired quality of service requirements. By matching Quality of Service attributes of service discovery requests and service descriptions, a client is able to distinguish services that match their requirements.
3) UDDI does not support stateless web service interaction where services are not responsible for storing session and/or state information generated due to service interactions.
4) In UDDI, service descriptions of a registry may be stale because of the volatile behavior of services. Services may come and go and service metadata can change frequently. To avoid out-of-date information in the registries, a monitoring scheme is required to keep the service-metadata up-to-date. [Mello]
5) UDDI introduces keyword-based retrieval mechanism. It is a poor service description and query, matchmaking process. The Semantic Web community provides solutions by

integrating ontologies, which in turn define rules on the concepts and relations, with UDDI. Unfortunately, this increases the complexity to the registry, and it is hard to come up with an ontology which everybody agrees and accepts as an standard. Main Semantic Web and UDDI research was originated from Cardiff Southampton and Univ. of Manchester. [Miles] [UDDIe]

6) UDDI is domain-independent and it does not provide domain-specific query capabilities such as performing geospatial queries on the metadata-catalog. That is why its adaptation in various specific domains such as GIS is slow. For instance, GIS community introduced their own registry, Web Registry Service as an alternative. Their solution is only designed for GIS domain and is not extendable. [Aktas]
7) UDDI is centralized and presents a single point of failure.

2. Analysis of Services in Particular Grids

2.1 Net-Centric Environment

NCE or NCOW (Net-Centric or Network Centric Operations and Warfare) is a “target architecture” designed by the US Department of Defense for all their future information technology – an area with a $30B annual budget. NCE is specified with voluminous documentation which in its full detail is not generally available. However the key features are available on the Internet from papers and talks including [DoD1] [Hayes04] [Krieger03] [Levitt05] and [Mayfield03]. There are also several NCE related organizations including the Association for Enterprise Integration with a Net-Centric services activities [AFEI]; the Industry oriented Network Centric Operations Industry Consortium [NCOIC]; NCOIC is managed by the Open Group which also has a Grid Enterprise Forum [GEF]; there is a broad based forum called the World Wide Consortium for the Grid [W2COG]. Note the latter focuses on the GiG or Global Information Grid which in DoD-speak is the infrastructure on which NCE is built. The NCE/GiG architecture is consistent with the principles of general DoD studies such as the 1998 C4ISR (Command, Control, Communications, Computer Intelligence, Surveillance, Reconnaissance) architecture document [C4ISRarch].

DoD divides NCE into three “views” termed Operational, System and Target Technical. The operational view describes functionality and requirements divided into a set of quite detailed activities. The systems view describes the computer architecture with its decomposition into services including those implementing the different activities of the operational view. The Target Technical View or TTY describes the technologies that could be used in the NCOW implementing the systems view subject to the requirements of the operational view. The TTY discusses the relevance of web services, Globus and the work of GGF. The NCE defines a set of 10 core services given in table 3 but there is no available discussion (other than [DoD1] and table 4 here) of the relationship of these to grid and web service specifications and how DoD could build their services on top of more general specifications. DoD recognizes the importance of different applications (called communities of interest) building on the same core (table 3) services. DoD also has application-specific metadata standards developed including DDMS XMSF XBML and C2IEDM (Command & Control Information Exchange Data Model) [DDMS, Tolk04A].

The NCOW is aimed at satisfying some broad principle required by DoD applications:

1. Increased Shared Awareness
2. Increased Speed of Command
3. Higher Tempo of Operations
4. Greater Lethality
5. Increased Survivability
6. Streamlined Combat Support
7. Effective Self-Synchronization

The net-centric environment is designed to realize four key features:

1. *Reach*: corresponding to rapid linkage between spatially separated entities
2. *Richness*: reflecting the interoperability of service architectures including the number of disparate services that can be brought to bear to solve time-critical problems.
3. *Agility*: reflects the adaptability enabled by transformations that can “plugged” in real-time into any system to enhance decision making
4. *Assurance*: corresponds to the ability to deliver superior performance as needed to combat heterogeneous adversaries.

DoD has a substantial discussion of the NCE data strategy with general goals (visible, accessible, understandable, trusted, interoperable, responsiveness to user needs, institutionalized) and some principle of use. The latter include “only handle information once” (OHIO), smart pull (as seen in subscription to pub-sub systems), posting data immediately before and after any processing, and support of shared spaces.

Table 4 relates DoD to Grid and Web service specifications where we note that collaboration is viewed as a key service by DoD while job submittal is not. The latter is essential to the “back-end” parts of DoD operations but not as critical to command and control or battlement management which is an essential focus of NCE. Note the latter application is similar to that needed for emergency response to civilian crises with the real-time linkage of commanders and first responders.

2.2 ServoGrid

http://www.servogrid.org		
Customers/Purpose: Support integration of large scale 3D simulations, data-mining (pattern recognition) and both archival and real-time data to predict earthquakes. Note this is different from NEESGrid that supports earthquake engineering (the consequences of earthquakes). The field is seeing an explosion in available data that will be dramatic if InSAR satellite launched. Further as earthquakes are a "phase transition" sensitive to unknown details of physical make-up of faults, the role and nature of simulations is still an active and controversial area and balance between very large accurate simulations and multiple exploratory runs is unclear.		
Area	**Service Name**	**Description**
FS3	Messaging Service	This is used to stream data in workflow fed by real-time sources. It is based on NaradaBrokering which can also be used in cases just involving archival data
FS4	Notification Service	This supplies alerts to users when filters (data-mining) detects features of interest
FS5 FS9	Workflow/Monitoring/Management Services	The HPSearch project uses HPSearch Web Services to execute JavaScript workflow descriptions. It has more recently been revised to support WS-Management and to support both workflow (where there are many alternatives) and system management (where there is less work). Management functions include life cycle of services and QoS for inter-service links
FS6	Authentication and Authorization	This uses capabilities built into portal. Note that simulations are typically performed on machines where user has accounts while data services are shared for read access
FS7	Registry and Discovery Services	We have built data model extensions to UDDI to support XPath queries over Geographical Information System capability.xml files. This is designed to replace OGC (Open Geospatial Consortium) Web registry service
FS8	Context Data Service	We store information gathered from users' interactions with the portal interface in a generic, recursively defined XML data structure. Typically we store input parameters and choices made by the user so that we can recover and reload these later. We also use this for monitoring remote workflows. We have devoted considerable effort into developing WS-Context to support the generalization of this initial simple service.
FS11	Portal	We use an OGCE based portal based on portlet architecture
FS11 FS15	Web Map Service	We built a Web Service version of this Open Geospatial Consortium specification. The WMS constructs images out of abstract feature descriptions.

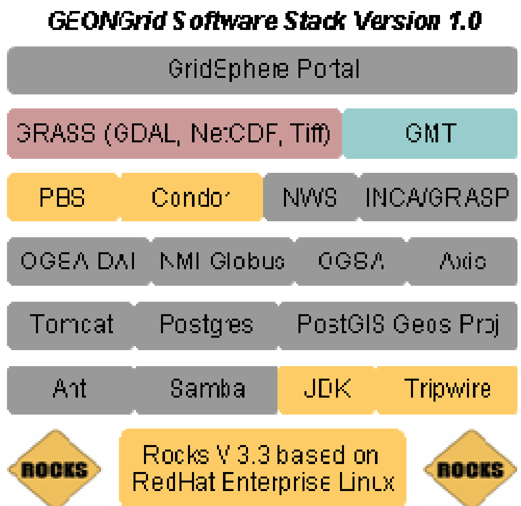
FS11 FS15	Scientific Plotting Services	We are developing Dislin-based scientific plotting services as a variation of our Web Map Service: for a given input service, we can generate a raster image (like a contour plot) which can be integrated with other scientific and GIS map plot images.
FS12A	Compute Access	We access specific job schedulers in straightforward ways as we do not need to link multiple compute jobs but rather single compute jobs on a user-chosen site with multiple services
FS13A	File Services	We built a file web service that could do uploads, downloads, and crossloads between different services. Clearly this supports specific operations such as file browsing, creation, deletion and copying.
FS13B	Sensor Grid Services	We are developing infrastructure to support streaming GPS signals and their successive filtering into different formats. This is built over NaradaBrokering (see messaging service). This does not use Web Services as such at present but the filters can be controlled by HPSearch services.
FS14B FS15.	QuakeTables Database Services	The USC QuakeTables fault database project includes a web service that allows you to search for Earthquake faults.
FS13A	Data Tables Web Service	We are developing a Web Service based on the National Virtual Observatory's VOTables XML format for tabular data. We see this as a useful general format for ASCII data produced by various application codes in SERVO and other projects.
FS14B FS15.	Application and Host Metadata Service	We have an Application and a Host Descriptor service based on XML schema descriptors. Portlet interfaces allow code administrators to make applications available through the browser.
FS14B FS15	Web Feature Service	We've built a Web Service version of this OGC standard. We've extended it to support data streaming for increased performance.
FS15	Specific Applications: Virtual California, Geofest, Park, RDAHMM ..	These can be all launched by a single Job Management service or by custom instances of this with metadata preset to a particular application
Key interfaces/standards/software Used		GML WFS WMS WSDL XML Schema with pull parser XPP SOAP with Axis 1.x UDDI WS-Context JSR-168 JDBC Servlets WS-Management VOTables in Research
Key interfaces/standards/software NOT Used (often just for historical reasons as project predated standard)		WS-Security JSDL WSRF BPEL OGSA-DAI

2.3 gLite 3.0.0

http://www.gLite.org		
Customers/Purpose: This Grid software is produced by combining the work of the EGEE with that of the LCG (LHC Computing Group). Curiously gLite 3.0.0 was obtained by converging gLite 1.5.0 (workload management system, logging and bookkeeping, R-GMA, VOMS and File Transfer Service) and LCG 2.7.0 (other components including those that duplicate gLite 1.5.0 functionality) http://lcg.web.cern.ch/LCG/LCGnews/LCGJunWeb-2006.pdf. LCG was the production system (based on European DataGrid EDG, GT2 and Condor) and gLIte 1.5.0 the "new system"; gLIte 3.0.0 is now the single supported system and is used in current round SC4 of LCG service challenges. Note gLite is the official CERN release and there are significant variants from the major LHC experiments including ATLAS http://www.gridpp.ac.uk/gridpp16/gridpp16_ATLAS.ppt, CMS http://www.gridpp.ac.uk/gridpp16/gridpp16_CMS.ppt, ALICE (using AliEn written partly in Perl) http://www.gridpp.ac.uk/gridpp16/gridpp16_ALICE.ppt and LHCb http://www.gridpp.ac.uk/gridpp16/gridpp16_LHCb.ppt. Further there is the related Open Science Grid USA effort whose software stack has a stronger Globus component. There again you will find variants of the core activity for each major LHC experiment. gLite would be termed in a compute-file grid in language of the Fox&Walker Gap analysis but is more normally termed a data grid. Note that few Grid and Web service standards are used and GT2 not GT4 components are used.		
Area	**Service Name**	**Description**
FS6	**Security**	https://edms.cern.ch/document/487004/ http://glite.web.cern.ch/glite/security/ Currently uses transport level security but will move to WS-Security
	Authentication	MyProxy LCAS (Local Centre Authorization Service) and LCMAPS (Local Credential Mapping Service) are the EGEE services that make the connection between GRID users certificates and local UNIX userids on the local site.
	Authorization	Uses VOMS
FS7	**Service Discovery**	Can access XML files (with service information), R-GMA or BDII information. BDII (Berkeley Database Information Index) is a sophisticated LDAP based system using two instances to improve performance (over original MDS).
FS12B	**Workload Management**	
	(Computing)	
FS12A	Computing Element	The Computing Element (CE) is the service representing a computing resource. Its main functionality is job management (job submission, job control, etc.). The CE may be used by a generic client: an end-user interacting directly with the Computing Element, or the Workload Manager, which submits a given job to an appropriate CE found

		by a matchmaking process. For job submission, the CE can work in push model (where the job is pushed to a CE for its execution) or pull model (where the CE is asking the Workload Management Service for jobs). Besides job management capabilities, a CE must also provide information describing itself. In the push model this information is published in the information Service, and it is used by the match making engine which matches available resources to queued jobs. In the pull model the CE information is embedded in a ``CE availability'' message, which is sent by the CE to a Workload Management Service. The matchmaker then uses this information to find a suitable job for the CE. Developing CREAM: Web service Computing Element
FS14A	Logging and Bookkeeping	Provides events defining status of a job throughout its life. L & B interfaces with R-GMA and uses proxies to improve performance
FS12A	Accounting DGAS	The accounting service accumulates information about the usage of Grid resources by the users and by groups of users, including Virtual Organizations as groups of users
FS12B	Workload Management WMS	Manage the whole job cycle including Condor capabilities, checkpointing, logging/bookkeeping and accounting.
FS13,14	**Data Management**	
FS13A	Storage Element	SRM used to virtualize storage devices with Posix-like Grid File Access Layer (GFAL) by LCG. There is also CASTOR (CERN Advanced Storage Manager)
FS13A	File & Replica Catalog	Fireman (gLite) and LFC (LHC File Catalog) supports caching of files to improve performance
FS13C	Data Movement	FTS Service controls File Transfer
FS14B	Metadata Catalog	AMGA from EGEE but it is expected that this will be very application (VO) dependent
FS14A	**Information and Monitoring**	R-GMA implements the GGF GMA architecture
		Using a relational database

2.4 GEON

http://www.geongrid.org/ **http://www.geongrid.org/communications/annual_reports/Annual_Report_2005_Final_Pub.pdf** 		
Customers/Purpose		NSF ITR addressing a broader range of GeoScience than SERVOGrid. The project is led by SDSC and has an emphasis on the Semantic Grid and Workflow. The annual report cited above has the best description of the architecture. The group is active in EarthScope and Geoinformatics communities. http://www.geongrid.org/related.html
Node Types		**Description**
Institutional (PoP) Node (following BIRN's deployment stragy)		OGSI (NMI Globus GT3.2); OGSA-DAI (no longer used); Postgres; PostGIS; NWS; INCA; Tripwire; Condor
GEON Central		Collection of useful software including applications managed by the distributed team.
FS6	GEON Certificate Authority	GAMA
FS12	GEON Compute Node	ROCKS
FS13,14	GEON Data Node	DB2; Oracle; SRB; Postgres; MySQL; ArcIMS. Note GEON software stack shown above does not illustrate this software.
Capability/Service		**Description**
FS5	Workflow	Kepler for LIDAR data processing and rock classification
FS6	Intrusion Monitoring	Tripwire for basic OS Files. chkrootkit for testing for rootkits on Linux servers
FS11	**Portal**	**A GridSphere based portal using portlets**
FS11 FS6	Registration and Login Portlets	GAMA (Grid Authentication and Account Management) is a complete GSI (Grid Security Inrastructure) credential management and integration solution tailored for use through Web portals or Web service-based clients. GAMA makes grid security as easy to use as any commercial web site while maintaining the security and delegation capabilities of GSI.

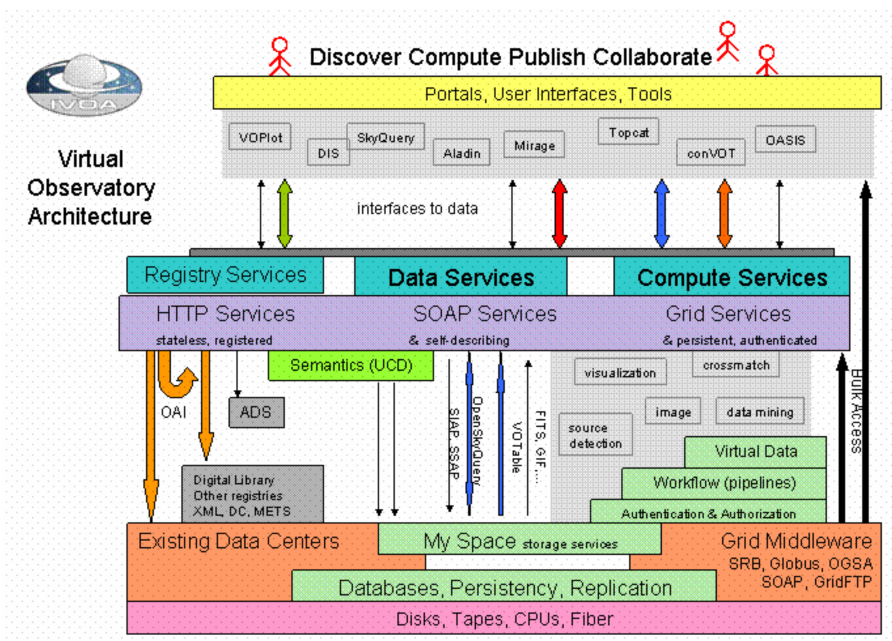
		GAMA consists of two components: a backend security service that provides secure management of credentials, and a front-end set of portlets that provide integration into web/grid portals. Gama is a general SDSC Grid technology. PostGRES handles user account information
FS11 FS9	System Administration Portlet	
FS11 FS12	Backup Portlet	
FS11 FS12	myGEON User Space	The GEON portal provides a private workspace called, myGEON, to store your work in progress. For example, you can store selected outputs from a GEONsearch, or store maps that you may have made using the GEON GIS mapping capability. You may log out and return to your work at myGEON.
FS11 FS14	Monitoring Portlet	
FS11 FS14	GEONSearch Portlet	Interfaces to GEON Metadata catalog
FS11 FS17	GEON Forum Portlet	
FS12	**Computing**	
FS12B	Cluster	ROCKS providing cluster management (customizable software configuration) integrated across the GEON sites by PoP nodes. See http://www.rocksclusters.org/rocks-doc/papers/ieee-cluster-2004/Rocks-Geon/Sacerdoti04GridManagement.pdf ROCKS is a general SDSC Cluster technology
FS12A	Job Scheduling	Globus and Condor used for scheduling Inter-clusters and to TeraGrid Sun Grid Engine and PBS used for Scheduling intra-Cluster
FS13,14	**Data**	
FS13A	Data Replication Service DRS	Used to replicate large data-sets using WSRF, RFT (Reliable File Transfer)
FS13C	Data Transport	GridFTP
FS14B	Hosted datasets	SRB used for large data; not metadata. SRB is a general SDSC dataset management system
FS14	**Information&Monitoring; IKW part of DIKW**	
FS14B FS15	Spatial Database	PostGIS extension to Postgres database which does not seem to be presented as a web service. This supports Geos and Proj tools
FS14B FS15	Geographical Information system GIS	ArcIMS and ArcGIS from ESRI; GRASS GIS is part of GEON Central distributed software repository; Generic Mapping Tools GMT from University of Hawaii. http://gmt.soest.hawaii.edu/
FS14A	System monitoring	INCA collects GEONGrid system status (inter-Cluster). This is a TeraGrid technology and supports GRASP Benchmarks Ganglia is packaged with ROCKS and used to monitor clusters
FS14B	Metadata Catalog	Based on Postgres frontended by GEON Search Portlet. The

		search interface accumulates search results in a “data integration cart” whose contents can be passed to other GEON services such as mapping and refined searches.
FS15	**Applications**	Parallel Finite-element code for time dependent 3D continental tectonics; SYNSEIS finite difference based generation of synthetic seismograms with GIS display; generation of ontologies and hosting of a range of Geoscience datasets; Kepler applications; visualization;
FS16	NWS Network Weather service	Monitor and forecast network systems properties

2.5 International Virtual Observatory Alliance IVOA

http://www.ivoa.net/

http://us-vo.org/summer-school/2005/proceedings/index.html is an excellent discussion of all components of the IVOA including technology and applicants. IVOA is an international standards and community building organization linking multiple (15 in May 2005) national projects including AstroGrid (UK) http://www.astrogrid.org and the National Virtual Observatory (USA) http://us-vo.org/. IVOA has 8 working groups and 4 community (interest) groups and sponsors general and interoperability meetings (See The IVOA in 2005: Assessment and Future Roadmap). The Working Groups follow the W3C model and are active in the areas of XML data format standards (VOTable), VO Resource Metadata, Universal Content Descriptions, Space-Time Coordinate Metadata, unified Data Access Layer standards for spectra and images, VO Resource Registries, VO Query Language, unified astronomical Data Models and Web Service technologies for the VO. The chairs of the Working Groups have also produced an overall architectural plan for an operational VO that identifies the critical areas for current and future development of standards and technologies. The work of the IVOA is an excellent complement to that other standards bodies and has a clear application focus.

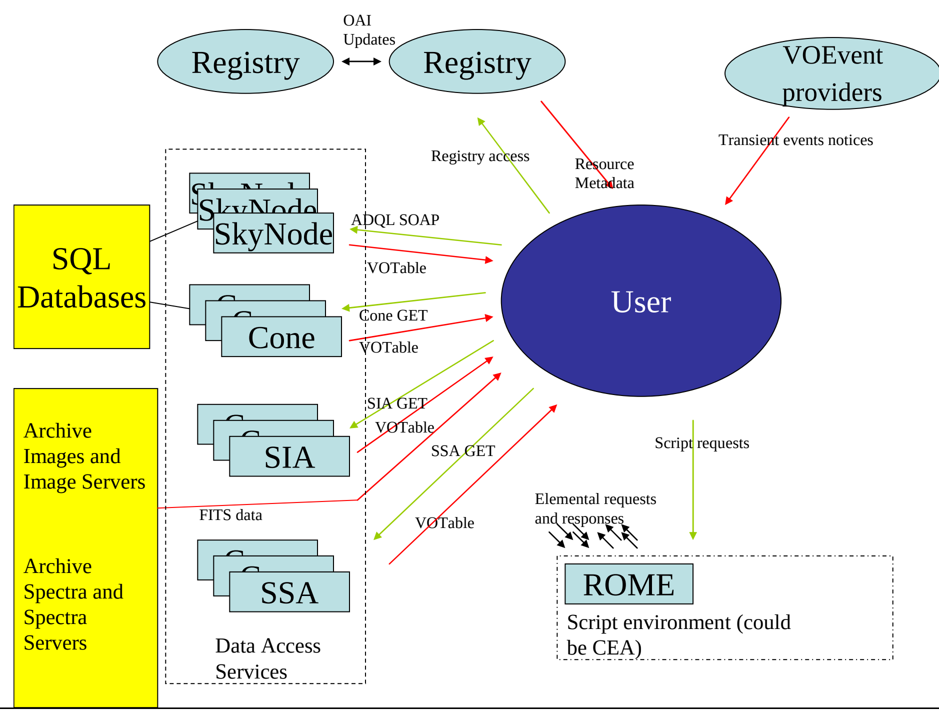


An Architecture for Scaling NVO Services to TeraGrid is a November 2005 report describing the integration of some US-NVO capabilities and the TeraGrid. This is an interesting architecture proposal that brings key data (SDSS and 2MASS) inside the TeraGrid moving away from the a full "take the computing to the data" model as seen in the earlier IVOA "data vision" seen in The Management, Storage, and Utilization of Astronomical Data in the 21st Century Version 1.00 IVOA Note 15 March 2004).

The IVOA architecture given in picture below (taken from Virtual Observatory Architecture Overview Version 1.0 IVOA Note 2004-06-14) is consistent with other applications. Note that My Space is now called VOSpace and is architected today as a central metadata store referencing (managing) distributed data repositories. The SRB has been used but not broadly as the International Alliance has not agreed to this choice; [NGAS](#) (Next Generation Archive System) from ESO (European Southern Observatory) has similar design goals to SRB. The articles VOSpace Architecture and National Virtual Observatory VOStore and VOSpace discuss in detail VOSpace, VOStore, NGAS and SRB.

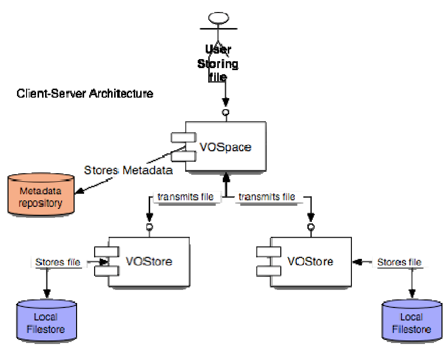
VOtable (see VOTable Format DefinitionVersion 1.1 IVOA Recommendation 11 August 2004) is an IVOA standard that specifies how table data should be represented in XML for both direct and indirect (URI) specifications. The data servers (SkyNodes) are expected to return their data in VOTable syntax. One can expect this approach to be useful across many scientific and engineering domains with a suite of Microsoft Excel style functions to manipulate the table data. Currently only VOPlot is available as a generic VOTable tool.

The diagram below shows a functional diagram focused on the data side of an IVOA Grid.



Node Types	**Description**
SkyNodes Database Server	http://us-vo.org/summer-school/2005/proceedings/presentations/sky_nodes.html A Database Server with an ADQL based SOAP interface returning VOTable based results. The VOQL group of IVOA is developing ADQL and Skynode specifications.
OpenSkyQuery Portal	http://openskyquery.net/Sky/skysite is a portal allowing access to multiple Skynodes

Capability/Service		**Description**
FS5 and FS12	Workflow JES/CEC/ CEA	JES (Job Execution System) is Astrogrid workflow engine that manages control flow and runs steps in a controlled asynchronous fashion. CEC (Common Execution Controller) manages step execution and data flow with CEA (Common Execution Architecture. This approach is AstroGrid

		(UK) Workflow Engine described in http://www.ivoa.net/Documents/Notes/AstrogridWorkflow/AstrogridWorkflow-20060227.pdf. It supports MySpace -- the forerunner of VOSpace. Interestingly this document references no other Grid Workflow activities – not even Taverna and Triana in UK where latter used in Astronomy (Ligo) http://us-vo.org/summer-school/2005/proceedings/presentations/VOData.ppt gives a broader discussion and the use of Pegasus as a workflow engine is described in http://www.isi.edu/~deelman/Pegasus/galmorph_sc03.pdf
FS7	Registry	http://us-vo.org/summer-school/2005/proceedings/presentations/dataservice.html http://nvo.stsci.edu/voregistry/index.aspx is a registry of Skynodes. It does not appear to be based on UDDI
FS12	Clarens Grid System	Coordination of compute and data services includes MySpace, workflow, batch job submission and access to data. An Architecture for Scaling NVO Services to TeraGrid describes the role of Clarens and compares it to GT4.
FS13A	Data Access VOStore	This abstracts the raw data access and the dividing line between it and VOSpace is still being debated. It does not cover meta-data (that's in VOSpace) and some discussion suggests it could be replaced by generic specifications like GridFTP or a simple application (file access only initially) of OGSA-DAI. Recently VOStore has been merged into VOSpace
FS14B	Data Management VOSpace	 This manages metadata and data-collections and sits between user (portal) and VOStore as in figure to right. Operationally VOSpace can function in near term as a SRB-NGAS-… interoperability layer.
FS14A and FS4	VOEvent	http://us-vo.org/summer-school/2005/proceedings/presentations/VOEvent_projects.ppt A publish-subscribe based event architecture for astronomical event notification.
Protocols		
FITS Flexible Image Transport System		FITS is the traditional standard used for the transport, analysis, and archival storage of scientific data sets in the astronomy community. It supports Multi-dimensional arrays: 1D spectra, 2D images, 3D+ data cubes; Tables containing rows and columns of information and Header keywords providing descriptive information about the data
VOTable		http://us-vo.org/summer-school/2005/proceedings/presentations/votables.html is an XML specification of a table result. The cell entries are typically URI's

	with a server component resolved in a VO Registry and a server specific ID resolved by the server. Note VOTable cells cannot themselves by new XML objects as this was considered to make parsing very hard. Rather one sets the UCD (Unified Content Descriptor) to ucd="meta.link.url", or ucd="meta.link.ivoa", or ucd="meta.text.xml" to specify that Cell contents are respectively a URL, a URI interpretable by a VO Registry or a chunk of XML.
ADQL	This is an enhancement of SQL for astronomy adding capabilities for fuzzy join and to select regions of the sky – this is analogous to spatial extensions of SQL used in GIS. This is intended to replace an older Cone Search capability
SIA Simple Image Access	This forms the data model with SSA for the IVOA. SIA data model is the familiar "astronomical image" which generally means a 2D sky projection with a data array that is logically a regular grid of pixels encoded as a FITS image, GIF/JPEG, etc. The SIA includes standardized dataset metadata such as provenance, Image geometry, Scale, Format, Position, Time of observation, Spectral bandpass and Access information
SSA	This forms the data model with SIA for the IVOA. A simple query POS, SIZE, FORMAT - like cone search or SIA possibly refined by spectral or time bandpass, etc. Most metadata in the query response is optional. On Data retrieval, the simple retrieval is URL-based returning a dataset "document" (VOTable, FITS, JPEG, etc.). In simplest case data could be wavelength, flux as text (for spectrum).
UCD	http://us-vo.org/summer-school/2005/proceedings/presentations/UCD.ppt Unified Content Descriptor to define astronomical type meta-data. Created at CDS Strasbourg (the first VO prototype) Harvested From 5000 tables, 20000 table columns To create ~450 UCD words Example pos.eq.ra means right ascension
Resource Metadata for IVOA VOResource	Resource Metadata for the Virtual Observatory Version 1.01 IVOA Recommendation 2004 April 26 describes in broad terms how various different metadata sources (Dublin Core, FITS, UCD etc.) should be integrated. A VOResource http://www.ivoa.net/twiki/bin/view/IVOA/VOResourceV010 is described by such metadata

2.6 CICC: The Chemical Informatics Grid

http://www.chembiogrid.org		
Customers/Purpose: The goal of this project is to support cluster analysis, data mining, and quantum simulation/first principles calculations on experimentally obtained data on small molecules with potential use in drug development. Small molecule data is gathered from NIH PubChem and DTP databases, with additional large molecule data available from service-wrapped databases such as the Varuna, Protein Data Bank, PDBBind, and MODB. NIH-funded High Throughput Screening centers are expected to deluge the PubChem database with assays of the next several years, making the automated organization and analysis of data essential. Data analysis applications are interestingly combined with text analysis applications applied to journal and technical articles to make a comprehensive scientific environment. Workflow is a key part of this project as it encodes scientific use cases. Many CICC services and the general approach are based on Cambridge University's WWMM project (http://wwmm.ch.cam.ac.uk/) led by Prof. Peter Murray Rust.		
Area	**Service Name**	**Description**
FS5 FS9	Workflow/Monitoring/Management Services	CICC uses Taverna from the UK e-Science Program/OMII. MyLEAD (from the LEAD project) is also being evaluated as a workflow environment.
FS6	Authentication and Authorization	Currently all services are openly available.
FS7	Registry and Discovery Services	Will inherit registry services through previously developed SERVOGrid work.
FS11	Portal and portlets	Use a JSR 168-based portal.
FS13A	File Services	No specialized service. URLs are used for naming files and simple remote download. Services developed previously for SERVOGrid can be used for uploads.
FS14B	NIH DTP Database Services	This provides access to the NIH Developmental Therapeutics Program (DTP)'s database of molecular screens against 60 cancer cell lines.
FS14B		This is a free service provided by the NIH and used by us.
FS14B	PubMed Search Service (in development)	PubMed provides a searchable online database of medical journal articles. CICC is developing harvesting services of the abstracts that can be combined with text analysis applications such as OSCAR3.
FS14B	SPRESI Services	CICC has developed clients/service proxies to the commercial SPRESI service (http://www.spresi.com/). SPRESI's scientific database houses extensive molecular and reaction data, as well as references and patents.
FS14B	VARUNA Database Service	This database contains molecular structure and more detailed information (such as force fields).
FS13A	VOTables Data Tables Web Service	CICC is developing a Web Service based on the

		National Virtual Observatory's VOTables XML format for tabular data.
FS15	Specific Applications: BCI, OpenEye, Varuna, AutoGEFF	CICC inherits job management services from SERVOGrid (including one based on Apache Ant) for managing the execution of both commercial and in-house developed high performance computing applications.
FS15, FS18	Condor and BirdBath	Examining the use of Condor and its SOAP interface (BirdBath) as a super-scheduler for Varuna applications on the TeraGrid.
FS15	ToxTree Service	This service wraps an algorithm for estimating toxic hazards in a particular compound. Useful in combination with other clustering programs in a workflow.
FS15 and related to FS8	OSCAR3 Service	This service (based on the OSCAR3 application developed by the WWMM group) performs text analysis on journal articles and other documents to extract (in XML) the chemistry-specific information (such as chemical compounds). SMILES may be automatically assigned to well-known compounds. This may be combined with more traditional database workflows and clustering algorithms.
FS15, FS18	CDK Services	CICC has developed a number of simple services based on the Chemistry Development Kit (CDK). These include similarity calculations, molecular descriptor calculations, fingerprint generators, 2D image generators, and 3D coordinate molecular generators.
FS15, FS18	OpenBabel Service	This service converts between various chemical formats (such as between InChI and SMILES).
FS15, FS18	InChIGoogle	For a given InChI (a string specification of a molecular structure), performs a Google search to return a page-ranked list of matches.
Key interfaces/standards/software Used		WSDL, SOAP (with Axis 1.x). CML, InChI, SMILES, Taverna SCUFl JSR-168 JDBC Servlets VOTables
Key interfaces/standards/software NOT Used (although these may be integrated in the future)		WS-Security, JSDL, WSRF, BPEL, OGSA-DAI

2.7 TeraGrid

http://www.teragrid.org		
Customers/Purpose: TeraGrid is the NSF flagship Grid. It encompasses resources at all the major NSF supercomputing centers. TeraGrid is coordinated through the Grid Infrastructure Group (GIG) at the University of Chicago, working in partnership with the Resource Provider sites: Indiana University, Oak Ridge National Laboratory, National Center for Supercomputing Applications, Pittsburgh Supercomputing Center, Purdue University, San Diego Supercomputer Center, Texas Advanced Computing Center, and University of Chicago/Argonne National Laboratory. TeraGrid has over 100 TFlops of computing resources and 15 petabytes of online and archival data storage. The TeraGrid service architecture is based on Globus GT4. Individual application groups are organized into Gateways. Currently they include SCEC Earthworks Project, Network for Computational Nanotechnology and nanoHUB, The Earth System Grid (ESG), the Virtual Laboratory for Earth and Planetary Materials (VLAB), the Biology and Biomedicine Science Gateway, the Open Life Sciences Gateway (OLSG), the Telescience Project, the Grid Analysis Environment (GAE), the Neutron Science Instrument Gateway and the TeraGrid Visualization Gateway. Most of these take the form of Portals and an associated additional set of services that are deployed on top of TeraGrid resources. The core services provided by TeraGrid are still evolving. Many of the higher level application services are provided by Gateways. As the gateways evolve, the plan is to release a set of common gateway services.		
Area	**Service Name**	**Description**
FS4	Notification Service	GT4 WS-Notification
FS5	Workflow/Monitoring/Management Services	All the major workflow tools are used in TeraGrid. The primary supported one is based on Condor-G, Dagman and GridShell.
FS6	Authentication and Authorization	GSI, Purse, MyProxy, VOMS are all used.
FS7	Registry and Discovery Services	GT4 Index Service.
FS8	System Metadata and State	GT4 is based on WSRF, which provides service level metadata and state.
FS11	Portal	The TeraGrid Gateway project uses OGCE, NanoHub, Clarens as needed for a particular gateway.
FS12	Compute Access	GT4 WS-Gram. TeraGrid does not have a grid-wide scheduler.
FS13	File Services	Global File System is provided by GPFS. Data movement: GridFTP, Globus RFT. Metadata services are Gateway specific.
FS14 FS15.	Information Services	TeraGrid provides services for knowledge about the software stack, standard services for user accounting are also being developed.

2.8 The LEAD Grid

http://portal.leadproject.org		
Customers/Purpose: The Linked Environments for Atmospheric Discovery project is about meso-scale storm prediction. It has a Grid that spans resources at seven universities (Oklahoma, Indiana, NCSA, Alabama Huntsville, UCAR, Millersville and UNC. While this is a stand-alone Grid, it is also an example of a TeraGrid Gateway. Both the LEAD Grid and TeraGrid are based on GT4 for many of the core services. However, on top of these, LEAD has a set of other essential services that support LEAD specific requirements. See [LEAD] for details.		
Area	**Service Name**	**Description**
FS4	Notification Service	WS-Eventing and GT4 WS-Notification
FS5	Workflow/Monitoring/Management Services	BPEL is the main workflow engine. This is supported by a "drop and drag" composition tool which is access via the portal. More details are provided in the section below on workflow.
FS6	Authentication and Authorization	GSI, Purse, MyProxy and a capability-based authorization system that embeds authorization tokens in each web service request.
FS7, FS14	Registry and Discovery Services	An information service that supports discovery of weather data as well as application service descriptions
FS8	System Metadata and State	GT4 is based on WSRF which provides service level metadata and state. Additional monitoring services are provided by UNC
FS11	Portal	The portal is based on a problem solving environment of their own design.
FS12	Compute Access	GT4 WS-Gram. Grid scheduling is provided by the VGrADS project. Individual applications are wrapped as webservices and managed by an application factory service.
FS13	File Services	.Data movement: GridFTP, Globus RFT. Weather data storage is provided by Unidata services. Metadata services are based on OGSA-DAI and comprise a personal metadata catalog for each user called MyLEAD

2.9 Naregi

http://www.naregi.org		
Customers/Purpose: Naregi is the National Research Grid Initiative of Japan. It contains a large middleware development project with a focus on applications in Molecular Science in Japan. There is a strong emphasis on nano-science. GT4 is the current framework, but they are targeting a much larger OGSA-based design.		
Area	**Service Name**	**Description**
FS4	Notification Service	GT4 WS-Notification and a Distributed Information Service.
FS5	Workflow/Monitoring/Management Services	BPEL is the main workflow engine. This is supported by a "drop and drag" composition tool which is access via the portal. More details are provided in the section below on workflow.
FS6	Authentication and Authorization	GSI seems to be the primary security model and
FS7, FS14	Registry and Discovery Services	A Distributed Information Service interoperates with the PSE
FS8	System Metadata and State	The Distributed Information Service also communicates with the Super scheduler and a network information service.
FS11	Portal	The portal is problem solving environment which is used to submit WFT Grid MPI jobs to the Super scheduler.
FS12	Compute Access	A Super Scheduler submits jobs to a set of Grid VMs. These manage reservations and co-allocation.
FS13	File Services	.Data movement: GridFTP, Globus RFT.

2.10 Additional Grid Efforts

The list of nine Grids above is not intended to be comprehensive. Many more exist. For example Fusion Grid is a collaboratory project from the Department of Energy Office of Science to support the fusion research community. Eco Grid is building an internet architecture for data management and analysis for ecological data. BIRN, the Biomedical Informatics Research Network is a grid that is focused on data analysis of biomedical imaging. The Laboratory for the Ocean Observatory Knowledge INtegration Grid (LOOKING) is a Grid built around data analysis of remote ocean sensors. The Earth Systems Grid is a DOE-funded project to build a Grid for climate modeling. The OGF maintains a list of an additional 20 Grids and see also appendix D of the NSF OCI Vision [OCIVision]. The similarity in the architecture of these Grids is remarkable. Each has resource discovery service, metadata catalog services, data and compute services, workflow tools and models and a security framework. Most use the WS-* core services and either GT4 or gLite as well as SRB or OGSA-DAI. In the case of Grid web service infrastructure, there is also more than GT4 and gLite. The OMII and Legion and WSRF.Net and ASKALON projects have all developed Grid service stacks, which each include a secure service container, authorization system, registry services, a notification system, a data system, and a job submission service. We are not aware of any application Grids that currently use these, but that may change.

3. e-Science Workflow Systems.

3.1. An Overview of Workflow Systems Challenges for e-Science.

In many ways, the requirements for service-based e-science workflows do not differ substantially from those of business workflows. The primary difference stems from the fact that enterprise workflows are about repetitive business processes and science is based on experiments. While experimentation has a significant repetitive component, the scientist is constantly altering the pattern of a workflow because that is where discoveries are made. Hence, ease of composition and editing, the ability to automatically log and record workflow enactments and the flexibility to incorporate new tools are all important features. But the ability to launch large-scale data analysis and simulation tasks from the desktop is emerging as the central feature and greatest challenge.

e-Science workflow tools have been built to address a wide spectrum of applications. At one end of the spectrum are tools that are designed to handle "desktop" tasks such as simple data analysis and visualization where the size of the data and computing requirements are relatively small. Included in this are frameworks that are designed to integrate a variety of desktop interactive tools as "plug-in" components. At the next level are workflow frameworks that run on the desktop, but allow the user to integrate remote services such as data and metadata directories so that information can be pulled into the desktop as part of the workflow execution. In some cases this remote service interact involves pushing a computational task to a specific remote service for execution. Finally there are those workflow systems that are designed to run large-scale e-science application on remote Grid resource. These systems need to support multiple concurrent user, deal with security, and run workflows that may take days to months to complete. The most advanced of these use a sophisticated layer of service too meet these requirements. In this report we focus on core Service Oriented Architecture components that make these systems work and we discuss possible standardization issues that confront them.

There are seven widely recognized Grid workflow projects. Many of these began life in the "desktop" workflow space, but they have evolved over time to address the large-scale e-science applications. The seven are:

1. Triana (Cardiff University [triana]). A graphical composition workflow system that began life as a desktop tool, but has evolved into a reasonable Grid-aware framework.
2. Kepler (SDSC and UC Davis [kepler]). Like Triana, Kepler began life as a desktop tool, but it has undergone a full evolution into a Grid-aware and service oriented system. Kepler is one of the most widely used of the e-Science workflow tools.
3. Taverna (University of Manchester [taverna]). A workflow system designed for the life-sciences. The first system to recognize the importance of data provenance and semantic grid issues.
4. Pegasus (University of Southern California, ISI [pegasus]). Based on DagMan, the Cactus workflow system, this project is one of the most developed for large-scale e-science applications.
5. ASKALON (University of Innsbruck [askalon]). This is a complete Grid framework for the construction of distributed workflows and their management and execution.
6. QoWL (University of Vienna [qowl]). A BPEL based e-Science workflow system that supports QoS as a first principle.
7. GPEL (Indiana University [gpel]). A BPEL based system that is designed for dynamic, adaptive large-scale e-science applications.

In addition to these seven there are another dozen or so workflow projects addressing e-Science. These include ICENI, Sedna, MOTEUR, BioOpera, Chimera, DiscoveryNet, Freefluo, GrindAnt, Karajan, Seige, JOpera and Teuta. Many of these systems have features that are as strong as the list of seven, so including them in a secondary list is not truly appropriate. However, our goal here is not to focus on the workflow systems, but rather the service architecture issues required to support e-Science.

In the case of e-Science there are a number of issues that are significant departures from the classic enterprise use-cases. These issues arise from the fact that many e-science workflows are based on compositions of large computational and data analysis tasks that must execute on remote supercomputing resources that are often organized as wide-area Grids. There are eight specific issue that must be addressed by e-Science workflows that are related to this execution model.

3.1.1. Abstraction - Hide the Grid.

Scientists want to get work done and they do not want to deal with the complexity of building workflows that expose details of the underlying Grid services or other middleware. The must be able to express their problem by composing application specific components in an easy to use, easy to re-use and easy to mody form. Their favorite model of programming the workflows is via a "drop-and-drag" graphical interface and they loath writing "programs" in XML. However, the visual programming model must be sufficiently powerful to address a wide range of conditions, exceptions, iteration and adaptive control.

3.1.2. Computation Virtualization

The advantage of a service-oriented architecture is that the activities in a workflow can be expressed in terms of application services. In scientific application these services are often based on executing a classical command line application on a parallel supercomputer. The scientist seldom cares about which supercomputer is used as long as the turn-around time for that step of the workflow is minimized. For example, suppose you have a large MPI program, called X, deployed on several different supercomputing hosts. If a service is made available that can execute program X given a complete description of all of the input parameters and files needed by X without specifically requiring information from the user/workflow designer about specific deployment details of the X program, then you have virtualized the computation. The problem of resource scheduling can then be pushed to the "X-service". Once it receives a request to execute program X, it can contact lower level Grid scheduling services that can mine information from the program inputs to determine the best resource on which X has been deployed and has the shortest job queue. The X-service can then stage the needed data files and submit the job to the resource.

There is one major drawback with the approach of having the application X-service negotiate with the resource broker services directly: it does not optimize overall workflow performance. For example, if two applications X and Y must be run in sequence and X produces many gigabytes of data that must be consumed by application Y, it may be best to find a single resource that can do both and minimize the data traffic. To do whole-workflow scheduling, the workflow enactment engine needs to be able to extract a compact description of the workflow and send it to an optimizing scheduler. The optimizing scheduler would return specific resource assignments to the enactment engine which can be passed to the specific application services.

3.1.3. Data Virtualization

The traditional scientific user spends a substantial amount of time managing remote data files and resources. However, the cost of storage is rapidly dropping to zero and Grid services are being deployed that can manage data and replicas of data automatically. All data products, including those that are intermediate results can be automatically cataloged and saved. If done correctly, the intermediate results can be reused in a related workflow or to restart a workflow that had a flaw in a downstream component. This requires the workflow enactment engine to be capable of recognizing these optimization possibilities. To accomplish this, the next three items (4, 5 and 6) are essential components of the workflow system.

3.1.4. Metadata Generation

One of the foundations of science is the requirement that experiments are repeatable and that all derived data products are traceable back to their sources. This has a profound impact on the workflow system and the application services. They must each be able to generate metadata that describes each data product (What were the inputs? When and where was it created? What version of the application code? What compiler was used to create the executable? What host did it run on? What OS version?) In addition, a detailed trace of the execution of the workflow itself (including the sequence of events, decision branch conditions, exceptions encountered) is needed. It is essential that the workflow system create this audit trail automatically, because the scientist will not have the patients to do this. It is also essential that a standard providence metadata schema be used.

3.1.5. Fault tolerance

e-Science workflows that run on distributed Grid resources must deal with fault handling at many levels. The lowest level that must be considered is dealing with the dynamic nature of the Grid. Resources come and go. Data movements fail because of congested networks or the lack of available local storage and machines crash or can be taken away because of higher priority tasks. The Workflow system must have mechanisms to track a failed step in a workflow, suspend the action and make a call to the resource broker to allocate new resources and then restart the workflow. Having data services that can retain the intermediate results generated by each workflow step is essential. It is also essential that the workflow enactment be re-startable from any point.

3.1.6. Dynamic, Adaptive workflows.

Faults are not the only dynamic behavior that workflow execution must contend with. In some cases the requirements of a workflow execution may involve responding to a stream of external events. For example, monitoring a sensor stream. Depending upon the events that are reported different actions may be required from the workflow. The implications for the workflow enactment engine are significant. First, workflows must have the ability to listen to event channels. Typically this means that they must be able to subscribe to events through a pub-sub system and allow actions in the workflow to be triggered by the arrival of specific events. Second, the workflows, once started, must be able to persist for very long periods. Another area of dynamic behavior is based on the case when there is a human in the loop. Few workflow systems allow a user to be an agent in a running workflow in a way that allows the user to change toe workflow on-the-fly. Finally, there is an important point concerning the way the workflow engine allows a workflow to dynamically evolve and adapt to changing requirements. This is an active area of research.

3.1.7. Compositional Orthogonality

Workflow systems that are based on composing service components must rely on having service components that are actually composable. For example a service that requires a particular type of input is of little value if there is not other components in your library that produce that type of output. What is typically done is to create data-type transforming components (called “shims”) that can be interposed between services with mismatched data product types. However, the problem can often be more complex. In some cases a change in a parameter type in one component can require a change in a parameter in another component way downstream in the system. In this case we have a type of lack of orthogonality between the components because the semantics of one depends upon the semantics of another. This is a very hard problem to detect and solve without extremely sophisticated and detailed knowledge about the semantics of each component service. None of the current systems provide a way to automate the generation of shims or to check for these deep semantic mismatches.

3.1.8. Security

If you wish to use large, expensive resource to do a computational experiment, you need the authorization to do so. The same is true of you wish to use data that may not be public. Most workflow systems that evolved from desktop tools do not have security infrastructure built-in. There are two standard parts to the problem. How do you authenticate the person running the workflow? Is that person authorized to use each of the services he or she has requested? The standard Grid security infrastructure provides a solution to the problem of authorization. A user with a valid Grid identity certificate can generate a proxy that can be used by the workflow engine to execute the workflow on behalf of the user. Authorization can be handled in two very similar ways: role-based attributes or capability-based tokens. In the capability based approach, the user's capabilities are loaded to the workflow engine along with the identity proxy certificate. When the workflow attempts to use a service it must pass the identity certificate along with the capability tokens to that service. If the capabilities match the services requirements, then the workflow can proceed.

3.2. *E-Science Workflow Services*

Given this list of eight areas of concern for large-scale e-science workflows, it is easy to enumerate the required services.

1. **Data Services** that can be easily invoked by the workflow engine or application services to manage data and replica storage for user workflows.
2. **User Metadata Catalogs** that can store the metadata associated with each workflow enactment including references to each data product as well as workflow execution traces.
3. **Job Scheduling and Resource Broker** services capable of scheduling both individual application services and whole workflows.
4. **Resource Registry Services** that can contain catalogs of available service components including complete semantic descriptions of each.
5. **Application Services and Factories** that can easily generate service wrappers for legacy application codes.
6. **Notification Services** that can be used in the wide-area for pub-sub event management.
7. **Security Services** for user authentication and authorization management.
8. **Monitoring Services** that can report to the Scheduling and Brokering system about the heath of all resources available.

To illustrate these services used in a large-scale e-Science workflow deployment we consider the NSF LEAD project. Figure 1 illustrates the core services used when a workflow is enacted. The user interacts with the system through the LEAD portal. The portal provides the basic point of user authentication and it is the primary access point to the workflow tools. From the portal (not shown in Figure 1), the user selects a workflow template and a input datasets. Together these form an "experiment" in LEAD and they are logged with the User's metadata catalog. Should the user decide to modify the workflow (or create one from scratch), a composition and monitoring tool is provided through the portal. This is a "drop-and-drag" graphical composer that compiles the user's workflow design into GPEL, an implementation of BPEL used in LEAD.

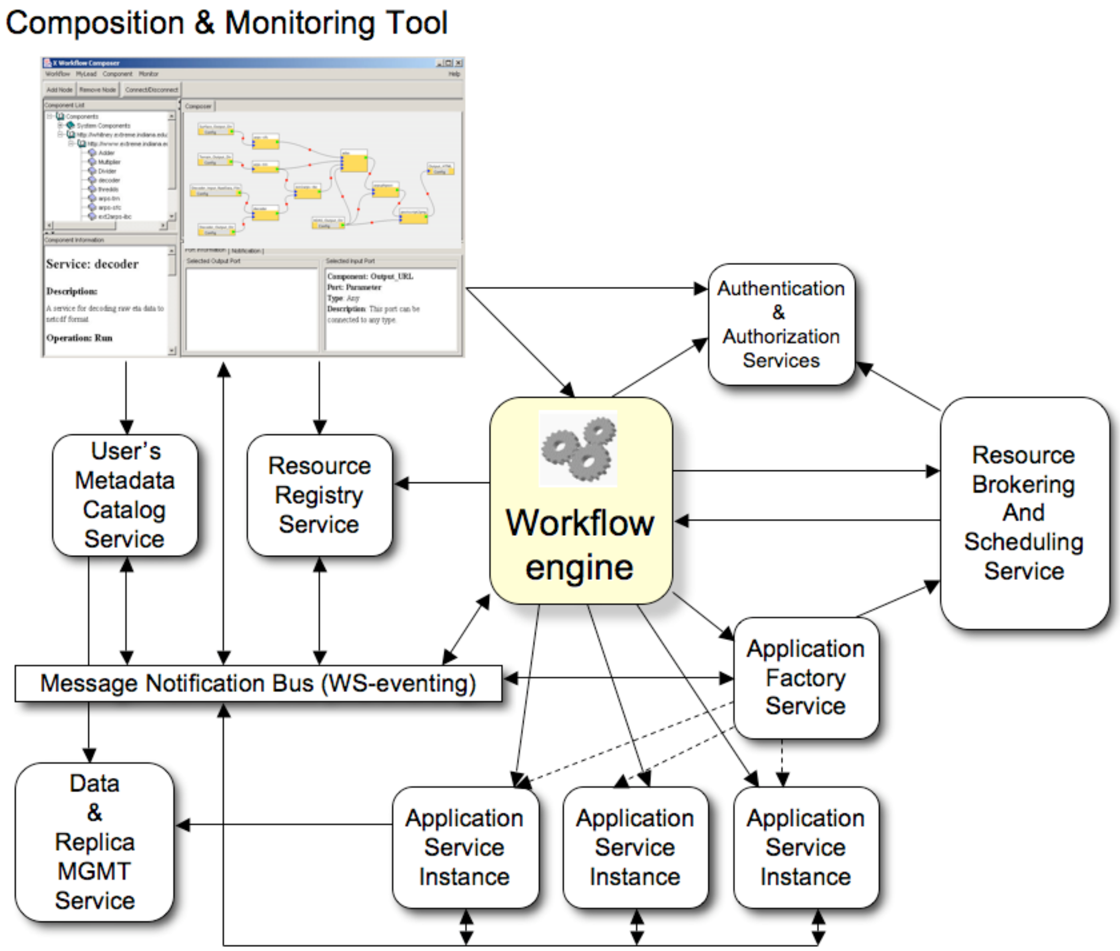


Figure 2. E-Science workflow services and lifecycle

The individual application service components are described in a resource registry service, which is shown to the user as a directory of tools which may be dropped on the composition pallet. (This is a model that is used in almost all visual programming systems.) The system must be able to authenticate the user and then verify that the user is authorized to use each of the application components that invoked by the workflow. This is done by both the portal and the workflow engine.

The basic template of the workflow is called the abstract workflow because it is not associated with any specific input parameters or specific binding of application services to real instances for an application on a specific host. Once the portal has associated the user's data selections with the abstract workflow it is sent to the workflow engine. The first thing the engine must do is to submit the workflow to a schedule planner and resource broker service. (In the LEAD project this is provided by the NSF VGrADS project led by Rice University.)

Once resources have been scheduled and allocated the workflow instance now has specific knowledge about the actual services it should use for each application node in the workflow. For example, in the abstract form of the workflow, there may be a point at which it states, "invoke the Weather Research Forecast (WRF) model". After scheduling it will say "invoke the WRF-service at 14:00 at the Pittsburgh Supercomputer Center". This specific WRF-service knows how to stage input files to the scratch directories of the machine, execute the WRF application and monitor its results. The application service will also generate metadata about data products and hand the data products to the underlying data management services. The application services also generate a stream of notification messages that document each step of the process (including any failures or exceptions). The metadata and enactment event histories are all stored in the user metadata catalog. These application service instances are managed by an Application Factory Service. The Factory creates specific application service instances as needed and each service instance can handle several hundred concurrent requests from client workflows. Finally, the workflow engine must monitor the notification bus to see if any resource has died or application service has failed. If this happens, the workflow engine must contact the scheduler and rerun the workflow instance from the last available active state. The workflow state is maintained in a database, so it is possible to resurrect running workflows even if the engine crashes.

4. Conclusions

All Grids that have exploited service architectures appear to report success in this regard and so we see that one should continue to emphasize and popularize Web or Grid service based Grids. On the other hand the situation with standards and specifications is much less clear. Only XML, SOAP and WSDL are broadly accepted but even here services with a REST protocol are likely to retain their popularity and should be accommodated. Looking at the other core areas FS3-11, the adoption is spotty but we do see agreement that the standards in notification, workflow, security, discovery, state, metadata, management, policy and portals are in the “right place”. There are significant differences in detail such that those between WS-Notification and WS-Eventing, and that between WS-Management and WSDM. Further the role of WSRF and the expression of state remains open and agreement here depends upon the next round of standards mergers. UDDI is the major specification in the area of service discovery but nearly all grids conclude it is inadequate and extend it in various incompatible ways as discussed in Sec 1.3. Grid Security has three components: authentication, authorization and privacy. Authentication is evolving to a combination of GSI, Kerberos and Shibbboleth, but X.509 and SAML based approaches dominate the field. Grid authorization remains unclear but as the infrastructure becomes more sophisticated (using databases for authorization as in VOMS and Permis for example) and as more of the WS-Security framework gets filled out, we can expect progress. . TeraGrid is working on a plan based on role-based authorization. This may be similar to the role-based model used in the Open Science Grid. A major problem with WS-Security is performance and this has slowed its adoption. Performance of WS-Security can be improved by not only better implementations but also by use of efficient representations conformant to the XML Infoset and not the traditional angle bracket representation. Further large scale experiment both academically and commercially will define better appropriate paradigms. It seems likely that some evolution of WS-security will provide an adequate security model.

It is not clear if there will be “one grid” or many but probably it doesn’t matter. One will be able to define a set of clear translation rules that map between the different models and it will be possible to perform these “on-the-fly” either in the basic service containers or in mediation services that act as SOAP intermediaries. These translations will be non trivial as for example translation from REST to SOAP WS-I+ service to WSRF requires movement and translation between header and body of message. Further one needs translation tables to perform such mappings. We anticipate growing adoption of standards in the core service specifications with perhaps consolidation around a “few stacks” between which translation is possible.

If we look at the higher-level services FS12-18, there is again success in using services but very little standardization of services except for GridFTP (which is outside web service framework) and a growing interest in JSDL for core job submission. OGSA-DAI has no competition as the database-Grid standard but many grids (as described in section 1) expose the business logic built around the database to the Grid so one can alternatively use well known non-Grid frameworks such as ODBC and JDBC to interact with the database. One intrinsic difficulty at this level is that in most cases, it is not obvious where the standard should be built. For example a user would not normally see JSDL or OGSA-DAI for computing and data but rather a “managed computing” or “managed data” interface. The user interface for managed computing could for example allow multiple jobs to be run in a parameter search and the managed data interface could integrate in a metadata catalog. In fact many computing grids are built in terms of software packages such as GT4, Condor, Unicore, OMII, SRM and SRB which provide the de facto high level standards. The new HPC OGSA Profile could provide some useful cluster computing standard.

Several metadata catalogs are available. One family is based on OGSA-DAI and another is based on the SRB MCAT. Standards are still far from available because application-level metadata schemas are emerging from many different communities. Many of these will converge, but what is needed is a framework for user metadata storage and search that is capable of dealing with an arbitrary application metadata schema.

One of the most interesting developments in this area is the VOSpace specification from IVOA which is a managed data interface. Note the needs of users and vendors are rather different. A vendor could use JSDL extensively inside a managed computing Grid whose users would just see a higher level interface. It is difficult to predict the future in this realm but we expect agreement to focus on software packages (or more generally workflows offering useful capabilities) with modest adoption of standards. This assessment partly reflects the immaturity of the field. Computing is relatively well understood and standards should be possible. The data area is critically important but needs more experimentation before standards can be developed. Examining the relevance of VOSpace and/or its extensions in other application areas seems a promising approach. The importance of “packages” or “Sub-Grids” such as SRB or Condor at this level has important implications for interoperability which is possibly best formulated at the Sub-Grid (system) rather than individual service interface.

The richness of models in the high level Grid service arena is illustrated by the OGF SAGA activity. Conventional wisdom is that you should probe a service with a SOAP message but this group is defining programming interfaces so remote access to services is possible from conventional (C, Java ..) programs. Clarification of the importance of this paradigm could be helpful.

There are many important standards outside the core service area used by Grids. One good example is JSR 168 which is purely a java standard but is extremely important for the portal community and all the standard java portal containers are based on JSR 168. However, JSR168 was not complete and version 2.0, termed JSR 286, is in the works. This will fix a bunch of shortcomings and it will integrate a second version of a WS-* spec that needed more work: WSRP or web services for remote portals (portlets). Some of our colleagues in our portal group OGCE are on this committee. Gannon and Pierce are editing a book on portals now so this is important to us. But because JSR168 (and 286) are very java specific we did not dwell on this topic in the report which tried to focus on the WS* and above level. Another class of specifications is illustrated by OPenDAP and the suite of OGC (Open Geospatial Consortium) specifications like GML, WMS and WFS. The latter are discussed under SERVOGrid in section 2 while the LEAD grid uses OPenDAP all the time because it is a standard in the atmospheric science community. It is a relatively low-level protocol that gives us some useful data-subsetting capabilities over HTTP for netCDF. It is a product from LEAD partner Unidata. It is useful to consider these “application-specific” standards while developing the core services because they must mesh together. For example OPenDAP needs to be supplemented with gridftp to make useful data transport. Again OGC has specified services such as discovery that clearly overlap the core service capabilities; we need to get powerful core services generally available and well established so that it is clear what the “applications” should assume and what they should add.

In the area of workflow systems, most projects rely on Globus for data services and remote job management. The area of Grid-level whole workflow scheduling is still a research topic. The same can be said of the area of Grid monitoring. In the latter case, there have been several attempts within GGF to standardize parts of this problem, but they have faltered because they were premature.

Resource registries are used heavily in workflow systems and they are also available in many forms. What is needed is a resource registry schema for workflow components that can be shared among workflow systems that provides in-depth semantics of workflow service components. There are enough similarities between application services used by different workflow systems that it is possible to make progress in this area. As an experiment for the LEAD project, we have made modest extensions to both Taverna and Kepler to allow them to execute some of simple workflows now orchestrated with BPEL. This proof-of-concept experiment shows that the same application services can be used with three of the best tools available. However, the next step is to define standard component semantics so that will allow reason about the correctness of workflow from any system.

A related issue to metadata catalogs is data provenance standards. Again, this is not a service definition issue, but rather a metadata schema standards issue. There is an effort that is starting in OGF to do this and it should be followed closely.

The application factory toolkits used by many different workflow systems vary widely in their capabilities. The one used in LEAD [gfac] is also now used in the RENCI BioPortal and at least one other project. It may be the most capable. As with many of the other core services, it is not an issue of standards as it is an issue of the best available technology.

Finally, another important part of workflow concerns events and notification. LEAD uses a combination of WS-Eventing and WS-Notification from GT4. It is expected that these two standards will be merged within the next few years. Both work and can be scaled to very large flows of messages. Other alternatives are systems like NaradaBrokering, which may be superior in many cases.

Standards in the Internet are based on the principle of "rough consensus and running code." The same concept seems to be emerging in the Grid workflow space. Good tools abound and they are maturing rapidly. There is already rough consensus in the life-sciences space that Taverna is system of choice. Kepler is very popular in other sciences, but BPEL based solutions may prove in the end to be more powerful. For the services the important part is not standardizing on service interfaces, but rather metadata schema for service components and workflow and their provenance. Investing in these schema standardization efforts may be the most productive way to achieve interoperability given that many different workflow systems will survive.

References

The detailed Grid descriptions in Section 2 have many embedded references (links). See also http://grids.ucs.indiana.edu/ptliupages/publications/gig/DODGridReferences.pdf and http://grids.ucs.indiana.edu/ptliupages/publications/gaprefs.doc

1. [AFEI] AFEI (Association for Enterprise Integration) NetCentric Enterprise Services Workshops http://www.afei.org/news/NCES_Workshops.cfm
2. [Aktas] Mehmet S. Aktas, Galip Aydin, Geoffrey C. Fox, Harshawardhan Gadgil, Marlon Pierce, Ahmet Sayar, Information Services for Grid/Web Service Oriented Architecture (SOA) based Geospatial Applications, Technical Report, June, 2005 http://grids.ucs.indiana.edu/ptliupages/publications/UDDI_Aktas_Final_Fix.pdf

3. [Askalon] Thomas Fahringer, Radu Prodan, Jurgen Hofer, Farrukh Nadeem, Francesco Nerieri, Stefan Podlipnig, Jun Qin, Duan Rubing, Mumtaz,Siddiqui, Hong-Linh Truong, Alex Villazon, Marek Wieczorek, ASKALON: A Development and Grid Computing Environment for Scientific Workflows
4. [C4ISRarch] Command, Control, Communications, Computer Intelligence Surveillance Reconnaissance (C4ISR) Core Architecture Data Model Version 2.0 http://www.fas.org/irp/program/core/fnlrprt.pdf
5. [CEE00] Collaborative Technology Development in the Air Force Research Laboratory Requirements for an Air Force Collaborative Enterprise Environment, AFRL January 2000 specification of CEE Architecture.
6. [CTSxx] See 2006 International Symposium on Collaborative Technologies and Systems CTS2006 May 14-17 2006 Las Vegas NV http://www.engr.udayton.edu/faculty/wsmari/cts06/
7. [DoD1] Geoffrey Fox, Alex Ho, Shrideep Pallickara, Marlon Pierce, Wenjun Wu *Grids for the GiG and Real Time Simulations*, Proceedings of Ninth IEEE International Symposium DS-RT 2005 on Distributed Simulation and Real Time Applications' Montreal October 10-12 2005 http://grids.ucs.indiana.edu/ptliupages/publications/gig
8. [DoD2] Geoffrey Fox *Collaboration and Community Grids* Special Session VI: Collaboration Grids and Community Networks in Proceedings of IEEE 2006 International Symposium on Collaborative Technologies and Systems CTS 2006 conference Las Vegas May 14-17 2006; IEEE Computer Society, Ed: Smari, Waleed & McQuay, William, pp 419-428. ISBN 0-9785699-0-3 http://grids.ucs.indiana.edu/ptliupages/publications/CTSMay06_cgc-03.pdf.
9. [DoD3] Geoffrey Fox, Alex Ho, Marlon Pierce *Appendix on Grid and Web Services* http://grids.ucs.indiana.edu/ptliupages/publications/gig/DODGridOpportunitiesAppendix.pdf.
10. [DDMS] Department of Defense Discovery Metadata Standard (DDMS) Version 1.2 http://diides.ncr.disa.mil/mdreg/user/DDMS.cfm
11. [GapAnalysis] Geoffrey Fox, David Walker *e-Science Gap Analysis* June 30 2003, http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/index.html
12. [GEF] The Grid Enterprise Forum http://www.opengroup.org/gesforum/
13. [GFAC] Gopi Kandaswamy, Dennis Gannon, Liang Fang, Yi Huang, Satoshi Shirasuna, Suresh Marru, "Building Web Services for Scientific Applications", IBM Journal of Research and Development, Vol 50, No. 2/3 March/May 2006
14. [GPEL] Aleksader Slominski, Adapting BPEL to Scientific Workflows, Workflows for eScience: Scientific Workflow for Grids, I. Taylar, E. Deelman, D. Gannon, M. Shields, eds., Springer-Verlag, 2007.
15. [Hayes04] Rick Hayes-Roth *Comments on NCOW RM 1.01*, October 19 2004 http://www.w2cog.org/documents/RHR_comments_re_NCOW_RM_1.01.doc
16. [Kepler] Deana D. Pennington, Dan Higgins, A. Townsend Peterson, Matthew B. Jones, Bertram Ludascher, Shawn Bowers. Ecological Niche Modeling Using the Kepler Workflow System, Workflows for eScience: Scientific Workflow for Grids, I. Taylar, E. Deelman, D. Gannon, M. Shields, eds., Springer-Verlag, 2007.
17. [KK] KnowledgeKinetics, Ball Aerospace enterprise collaboration environment in CEE architecture http://www.ball.com/aerospace/k2_home.html

18. [Krieger03] Mike Krieger Director, Information Management DASD(DCIO), OASD(NII) *NCES Net-Centric Enterprise Services* The Open Group Conference Washington DC 20-24 October 2003 http://www.opengroup.org/public/member/proceedings/q403/krieger.pdf
19. [LEAD] K. Droegemeier, D. Gannon, D. Reed, B. Plale, J. Alameda, T. Baltzer, K. Brewster, R. Clark, B. Domenico, S. Graves, E. Joseph, D. Murray, R. Ramachandran, M. Ramamurthy, L. Ramakkrisshnan, J. Rushing, D. Weber, R. Wilhelmson, A. Wilson, M. Xue, S. Yalda, "Service-Oriented Environments for Dynamically Interacting with Mesoscale Weather", CiSE, Computing in Science & Engineering – November 2005, vol. 7, no. 6, pp. 12-29
20. [Levitt05] Bill Levitt, NCOW RM Development Group *Update on Target Technical View - Emerging Net-Centric Standards - NCOW Reference Model v1.1* The Open Group Conference January 25, 2005, San Francisco http://www.opengroup.org/gesforum/uploads/40/6574/NCOW_TTV_V1.1_Open_Group.ppt
21. [Mayfield03] Terry Mayfield IDA *Net Centric Operations & Warfare Reference Model(Version 1.0)* National Defense Industry Association 3 October 2003 http://www.afei.org/pdf/ncow/Mayfield.pdf
22. [Mello] Adrian Mello, "Breathing new life into UDDI", Tech Report, ZDNET.com, 2002 available at http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2872349,00.html
23. [Miles] Personalized Grid Service Discovery. Miles, S., Papay, J., Dialani, V., Luck, M., Decker, K., Payne, T., and Moreau, L. Nineteenth Annual UK Performance Engineering Workshop (UKPEW'03), University of Warwick, Coventry, England, 2003.
24. [NCOIC] NCOIC Network Centric Operations Industry Consortium http://www.ncoic.org/
25. [OCIVision] NSF Cyberinfrastructure Council NSF'S CYBERINFRASTRUCTURE VISION FOR 21ST CENTURY DISCOVERY http://www.nsf.gov/od/oci/ci-v7.pdf
26. [Pegasus] Ewa Deelman, Gaurang Mehta, Gurmeet Singh, Mei-Hui Su, Karan Vahi, Pegasus: Mapping Large-Scale Workflows to Distributed Resources, Workflows for eScience: Scientific Workflow for Grids, I. Taylar, E. Deelman, D. Gannon, M. Shields, eds., Springer-Verlag, 2007.
27. [PicklesItaly] S. Pickles *Collaboration Grids: costs, benefits and scaling laws*, HPC2006 July 6 2006 Cetraro Italy http://www.hpcc.unical.it/hpc2006/prsnts/pickles.ppt
28. [Taverna] Tom Oinn, Peter Li , Douglas B. Kell , Carole Goble, Antoon Goderis, Mark Greenwood, Duncan Hull, Robert Stevens, Daniele Turi and Jun Zhao, Taverna / myGrid: aligning a workflow system with the life sciences community, Workflows for eScience: Scientific Workflow for Grids, I. Taylar, E. Deelman, D. Gannon, M. Shields, eds., Springer-Verlag, 2007.
29. [Tolk04A] Andreas Tolk, *XML Mediation Services utilizing Model Based Data Management*, 2004 Winter Simulation Conference, SCS, Arlington, VA, December 2004 http://www.vmasc.odu.edu/publications/Tolk/tolka42424i.pdf
30. [Triana] Ian Taylor, Mathew Shields, Ian Wang and Andrew Harrison, The Triana Workflow Environment: Architecture and Applications, Workflows for eScience: Scientific Workflow for Grids, I. Taylar, E. Deelman, D. Gannon, M. Shields, eds., Springer-Verlag, 2007
31. [Qowl] Ivona Brandic, Sabri Pllana and Siegfried Benkner, High-level Composition of QoS-aware Grid Workflows: An Approach that Considers Location Affinity, Proceedings of WORKS06, Paris, 2006.

32. [UDDI] Bellwood, T., Clement, L., and von Riegen, C. (eds) (2003), UDDI Version 3.0.1: UDDI Spec Technical Committee Specification. Available from http://uddi.org/pubs/uddi-v3.0.1-20031014.htm
33. [UDDIe] Ali ShaikhAli, Omer Rana, Rashid Al-Ali and David W. Walker, UDDIe: An Extended Registry for Web Services. Proceedings of the Service Oriented Computing: Models, Architectures and Applications, SAINT-2003 IEEE Computer Society Press. Orlando Florida, USA, January 2003