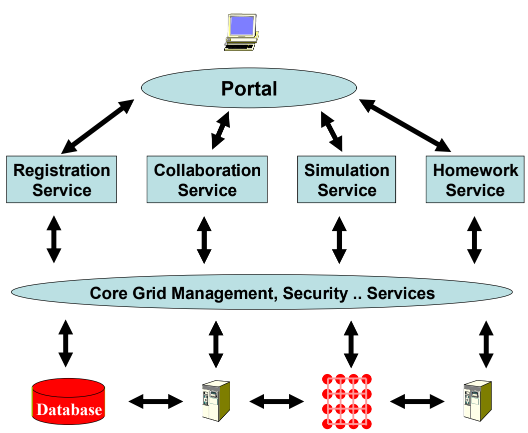
Education and the Enterprise with the Grid

Geoffrey Fox
Computer Science, Informatics and Physics, Indiana University
Community Grid Computing Laboratory,
501 N Morton Suite 224, Bloomington IN 47404
gcf@indiana.edu

1 Introduction

In this short article, we aim to describe the relevance of Grids in education. As in fact information technology for education builds on that for any organization, we first discuss the implication of Grids and web services for any organization – we call this an Enterprise to stress the importance of the goal and application specific features in any Grid deployment. The discussion of the importance of Grids for virtual organizations in Chapter 6 already implies its importance in education where our organization involves learners, teachers and other stakeholders such as parents and employers. We describe in section 2, the role of Web services and their hierarchical construction in terms of generic capabilities and applications of increasing specialization. In section 3, we summarize this in terms of a Web service implementation strategy for a hypothetical enterprise. Finally in section 4, we describe education grids pointing out the differences and similarities to general enterprises. We stress Web service issues, as these require the most substantial enterprise specific investment for they embody the particular objects and functionalities characteristic of each domain. The Grid provides the infrastructure on which to build the various Web service implementations. Deploying Grid infrastructure will get easier as commercial support grows and the heroic efforts described in chapter 5 are packaged properly.



Fig, 1: Typical Grid (Education) Enterprise Architecture

One will of course still have to worry about needed resources – computers, data storage and networks. On these one will install "core" Grid software infrastructure whose many components and approaches are described in part B of this book. This is the bottom two layers of Figure 1. On top of this, one will need several services – some could be generic like collaboration and others very specific to the enterprise – such as a homework submission service in education. It would be wonderful if there was a clear hierarchy but this will only be approximate with services connected say with "science", "people" and "education" not having a clear hierarchical relationship. Rather we will have a complex network of services with an approximate hierarchy; core services at the bottom of Figure 1 and portals handling user-facing service ports at the top (chapter 18). In this chapter we focus on the filling of the portal-core Grid sandwich which we discuss below for first general enterprises and then education. Although we are not certain as to exactly what depends on what, we are certain that we have a service model and that the interfaces are defined in XML. This we can start to address today without worrying too much about how technology evolves. For this reason, we discuss in most detail how Web services can be deployed in particular domains.

2 Web Service Paradigm for the Enterprise

We suppose that Web services will be developed for a wide variety of applications ("all of them") and that there will a corresponding suite of XML schema describing the object and services associated with each application. The net result will be a hierarchical structure of information and services. This has been described earlier in especially chapters 14 to 19 of this book on the role of data and the Grid. Let us imagine we are the chief information officer of some enterprise and wish to adopt a uniform Grid and Web service enabled view of our information technology environment shown in Figure 2. We would of course adopt a service architecture and define this with XML Schema for both our data structures and the functions (Web services) that operate on them,. We assume this will eventually be set up hierarchically as sketched in fig. 2. Our application would define its schema and, this would be used on top of other standards for example those of computing and databases as shown on the top of Figure 2. These specific Grid-wide application standards would themselves be built on general Grid, Web and Internet (IP) protocols.

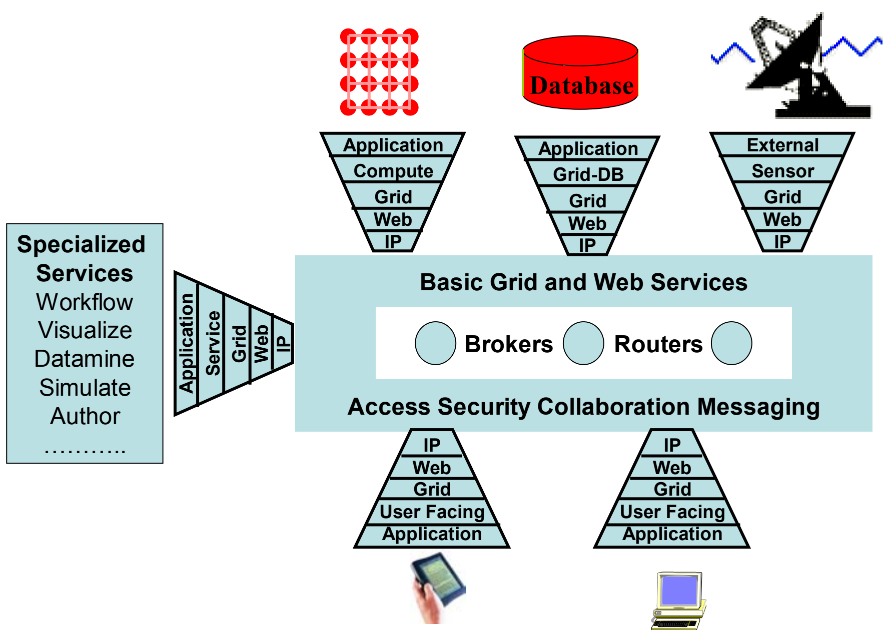


Fig. 2: Possible View of Enterprise Grid for a particular application showing hierarchical information structure with at the top, a parallel computer, database and sensor linked to the Grid. The left part of diagram lists important services with user interface devices at the bottom. The hierarchical interface is shown at top and bottom

Even our application could itself be composite and built up hierarchically internally – suppose our enterprise was a physics department of a university; then the "application schema" could involve a mixture of those for *physics* (extending a Schema for *science*) *research* and *education*. It could also involve Schema specific to the home university. As we will see later, the *education* schema itself is composite. Notice that this hierarchical information model is projected to the user through application related content rendered to clients through user facing ports on the Web service. As chief information officer, we would certainly try to ensure that our entire system, respected this single albeit complex representation. Figure 2 illustrates that there will be some places we need foreign (external) formats. At the top right, we assume that we have a scientific instrument on our grid and this has some distinct external specification. We imagine that the Grid community has defined some sort of *sensor* schema into which we can add the instrument. We now build a custom conversion web service that maps this device into the common data and service model of our grid. This process allows us to use the same *application* schema for all services and so build an integrated grid.

Another example could be a grid servicing a group of hospitals where we have devised a single specification of all medical, administrative and patient data. This is the interoperability language of the healthcare grid linking the hospitals together but realistically many hospitals in the chain would have their own (pre-existing) information systems with disparate data representations. In designing our grid we would represent each hospital's legacy system as an external extension to a base health care schema and then design mapping (web) services that converted all to the common interoperable representation. This discussion is meant to illustrate that building an enterprise (application) specific grid involves study of the different current representations of related systems and where possible adopting an hierarchical architecture based on more general applications.

The hierarchy of Web services is explored in tables 1 to 6. The last three tables describe application of Web services to science, education and research and will be discussed later in section 4. Here we want to describe briefly generic (tables 1 and 2), commodity and business services (table 3). We want to make two important points here

- All electronic processes will be implemented as Grid or Web services
- The processes will use objects described by XML defined by Schema agreed by particular organizations. Of course the Web services are XML described methods (functions) which input and output information specified by the XML application object specifications.

Table 1: Some Basic Grid Technology Services

Security Services	Authorization, authentication, privacy
Scheduling	Advance reservations, resource co-scheduling
Data Services	Data object name-space management, file staging, data stream management, caching (replication)
Database Service	Relational, Object and XML databases
User Services	Trouble tickets, problem resolution
Application Management Services	Application factories [32], lifetime, tracking, performance analysis,
Autonomy and Monitoring Service	Keep-alive meta-services. See [34]
Information Service	Manage service meta-data including service discovery [40]
Composition Service	Compose multiple Web services into a single service
Messaging Service	Manage linkage of Grid and web services [42]

Note that Web services are combined to form other web services. All the high level examples, we discuss here and give in the tables are really composites of many different Web services. In fact this composition is an active area of research these days [41] [1] and is one service in table 1. Actually deciding on the grain size of Web services will be important in all areas; if the Services are too small, communication overhead between services could be large; if the services are too large, modularity will be decreased and it will be hard to maintain interoperability.

Table 2: General Application level Services

Portal	Customization and Aggregation
People Collaboration	Access Grid - Desktop Audio-Video
Resource Collaboration	Document Sharing (WebDAV, Lotus Notes, P2P), News groups, channels, instant messenger, whiteboards, annotation systems. Virtual Organization Technology [31]
Decision Making Services	Surveys, consensus, group mediation
Knowledge Discovery Service	Data mining, indexes (directory based or unstructured), metadata indices, digital library services. Semantic Grid
Workflow Services	Support flow of information (approval) through some process, secure authentication of this flow. Planning and documentation
Universal Access	From PDA/Phone to disabilities; language translation

Table 3: Some Commodity and Business Applications

News & Entertainment	The Web
Video-on-Demand	Multi-media delivery
Copyright	The issues that troubled Napster done acceptably
Authoring Services	Multi-fragment pages, Charts, Multimedia
Voting, Survey Service	Political and Product Review
Advertising Service	Marketing as a Web service
e-Commerce	Payment, Digital Cash, Contracting; electronic marketplaces (portals)
Catalogs	As used in online sites like Amazon
Human Resources; and ERM	Uses privacy, security services; performance, references; Employee Relationship Management ERM as a Web Service
Enterprise Resource Planning ERP	Manage internal operations of an Enterprise
Customer Relationship Management CRM	B2C (Business to Customer) as a web service. Call Centers, integration of reseller and customer service Web services.
SFA Sales Force Automation	Manage sales and customer relationship; contacts, training
Supply Chain Management SCM	Typical B2B (Business to Business) Web services; also partner relationship management, collaborative product commerce (CPC) etc.
Health care	Patient and other hospital records, medical instruments, remote monitoring, telemedicine

Table 1 contains services that have been discussed in detail in part B of this book, chapters 6 to 19. These are the services creating the Grid environment from core capabilities such as security [30] and scheduling [33] to those that allow databases to be mounted as a Grid service [35][36][37]. The table 2 services have also been largely discussed in the book and consist of core capabilities at the “application Web service level”. Collaboration is the sharing of Web services as described in [39] [40], while portals are extensively discussed in part C of the book, chapters 20 to 34. Universal access covers the customization of user interactions for different clients coping with physical capabilities of user and nature of network and client device. The same user-facing ports of web services drive all clients with customization using the universal access service [39]. Workflow builds on the composition service of table 1 but has additional process and administrative function. Moving from data to information and then knowledge is critical as has been stressed in [37] and [38] and various data mining and meta-data tools will be developed to support this. The Semantic Grid is a critical concept [38] capturing the knowledge related services.

Table 3 illustrates broad-based application services that are developed to support consumers and business. The Web itself is of course a critical service providing “web pages” on demand. This is being extended with video-on-demand or high quality multi-media delivery; given the controversy that music downloading has caused we can expect copyright monitoring to be packaged as a service. Authoring – using Microsoft Word (and of course other packages such as Star Office, Macromedia and Adobe) – is an interesting Web Service; implementing this will make it a lot easier to share documents (discussed in section 4) and build composite web sites consisting of many fragments. We will derive our curriculum preparation service for education by extending this core authoring service. Voting, polling and advertising are commodity capabilities naturally implemented as Web services. The areas of internal enterprise management (ERP), B2B(Business to Business) and B2C (Business to Consumer) are being re-implemented as Web services today. Initially this will involve rehosting databases from companies like Oracle, PeopleSoft, SAP, and Sybase as Grid services without necessarily much change. However the new Grid architectures can lead to profound changes as Web services allow richer object structures (XML and not relational tables) and most importantly interoperability. This will allow tools like security and collaboration to be universally applied and the different Web services to be linked in complex dynamic value chains. The fault tolerance and self-organization (autonomy) of the Grid will lead to more robust powerful environments.

3 Implementing Web Services

We have learnt that gradually everything will become a Web service and both objects and functions will be specified in XML. What does this mean for our harried chief information officer or CIO that we introduced in the last section? Clearly the CIO needs to rethink their environment as a Grid of Web services. All data, information and knowledge must be specified in XML and the services built on top of them in WSDL [4]. The CIO will study the building blocks and related applications as exemplified in tables 1 to 3. This will lead each enterprise to define two key specifications – YEIF (Your Enterprise Internal framework) and YEEF (Your Enterprise External Framework). These could be essentially identical to those used in similar enterprises or very different if our CIO has a quite distinct organization. The YEEF is used to interface outside or legacy systems to the enterprise grid – we gave examples of a physics sensor or a legacy healthcare database when discussing Figure 2 above. Internally the enterprise grid will use the customized XML based framework YEIF. When you accept bids for new software components, the vendor would be responsible for supporting YEIF. This would be defined by a set of Schemas placed on a (secure) Web resource and always referenced by URI (universal Resource Identifier). YEIF would inevitably have multiple versions and the support software would need to understand any mappings needed between these. There would be an XML database managing this schema repository which would need to store rich semantic information as discussed in chapters 17 and 19; the UDDI effort [2] is trying to define such an enhanced schema storage but much work needs to be done here. Probably software referencing data structures defined by YEIF would not just be written in the programmer's or CIO's favorite programming model – rather the data structures would be generated automatically from the XML specification using technology like Castor [3]. This suggests new programming paradigms where data structures and method interfaces are defined in XML and control logic in traditional languages. Note that although interfaces are specified in XML, they certainly need not be implemented in this way. For instance we can use the binding feature of WSDL [4] to indicate that different, perhaps higher performance protocols are used that preserve the XML specification but have a more efficient implementation than SOAP [5].

The Web service approach gains interoperability from greater use of standards. Thus our CIO must be aware of and perhaps involved in the community processes defining Web service relevant standards for the application areas of importance to the Enterprise.

4 Education as a Web Service

We will simplify our discussion and only consider education for science and engineering. It will be straightforward to generalize to any curricula area but this is the author's expertise. Further science and engineering has extensive existing experience on use of electronic information, instruments and computer simulations in education. Figure 3 extends the generic environment of Figure 2 to education.

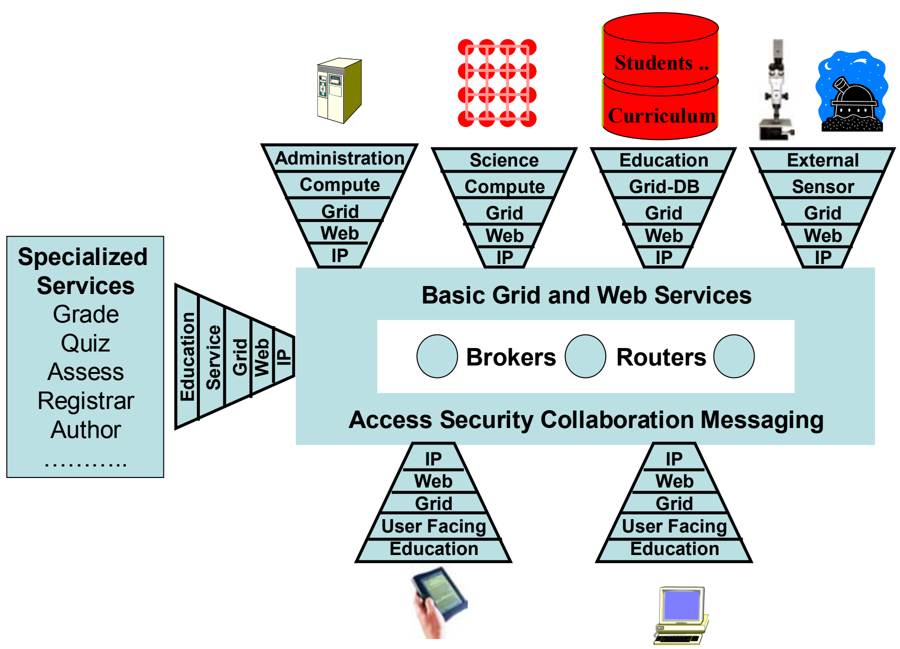


Fig. 3: A view of Grid in Education illustrating typical capabilities

Currently one uses rather specialized learning (content) management systems as the heart of a sophisticated learning environment. Such systems will be reworked to use generic Web services as much as possible. There will be specialized learning objects but functions like authoring and meta-data management will use the generic services of tables 1-3. Already this field has an excellent XML based object model through the work of the IMS [6] and ADL [7] initiatives. These have technical problems – they were not designed to a Grid or even Web architecture but rather to the client-server world of yesteryear. Further they are designed to be standalone rather than extending existing Service and XML –based data structures. These deficiencies are straightforward to address and these standards give us a clear object model for learning. We currently do not have services defined and these must be added – hopefully these national consortia will recognize this for it will not be difficult if they adopt the Grid architecture.

Table 4: Science and Engineering Generic Services

Authoring and Rendering Specialized to Science	Storage Rendering and Authoring of Mathematics, scientific whiteboards, n dimensional (n=2,3) data support, visualization, Geographical Information Systems, Virtual worlds
Discipline wide Capabilities as Network Services	Generic Mathematics (algebra, statistics, optimization, differential equation solution, image processing)
Sensor Services	Support general instruments (time series)
Tenure Evaluation	Shared with all scholarly fields; references. Specialization of generic Human Resources service

We assume a similar approach to that described in the last two sections for a typical Enterprise. Education is a very natural and important application of Grid technologies. Although "Education Grids" are not particularly common, the ideas underlie many of the efforts in distance education such as those of the Department of the Defense with ADL (Advanced Distributed Learning ADL [7]) and my own research in this area [8] [9]. The Biology Workbench from NCSA and now SDSC [10] is a particular good early example of an Education and Research Grid for science. There are several other examples developed by NSF's EOT-PACI program [11]. Grids offer support of virtual organizations – and clearly the network of learners, teachers, mentors, parents, administrators that is education form an interesting heterogeneous distributed virtual organization. Education has some special features of relevance to Grids. On the good (easy) side, education does not typically stress performance, as files tend to be of modest size for even if one uses simulations to illustrate educational issues, these need not be of the highest resolution. Important time scales are illustrated by the 30 milliseconds typical of an audio-video frame. Although this time scale is not in the microsecond range needed by parallel computing, quality of service is critical in education. Learning is hard and poor information delivery such as any distortion of audio packets (which only need some 10 kilobit/second bandwidth) will render the learning environment unacceptable [12]. This is particularly relevant for so-called synchronous learning where participants are linked in real-time in an interactive session – such as a delivery of a class over the Internet with teacher and students in different locations. Although this case in important and should be supported by an education Grid, most technologies in this book are aimed at asynchronous learning. Resources (curriculum – lectures, homework and quizzes) are shared but not accessed simultaneously. Probably in terms of student time, asynchronous learning is nearly always dominant but in many education paradigms, the synchronous case is also essential and a distinctive requirement of an education Grid. One interesting feature of an education Grid is the richness of the (meta-) data illustrated by the properties defined by IMS and ADL and by the many special Web services in table 6. Consistent with the lack of emphasis on performance, education does not have individually huge data blocks but rather a myriad (as many students) of very rich XML structures. We can expect XML's natural support of complex objects to be more important in education than some other enterprises.

As mentioned we will discuss an education Grid for science and engineering fields and adopt the hierarchical model used in section 2. First we assume that science and engineering will be implemented as Web services and in table 4, give a few simple examples. Note we will for brevity drop engineering in the following text and discuss science even though engineering has essentially identical considerations. Table 5 specializes to research, which corresponds to e-science as discussed in several places in this book – especially chapters 1, 7, 35 and 36. Table 6 lists a set of critical education Web services, which are applicable in many fields. Table 4 notes the special importance of mathematics and support of the natural topologies of science – two and three-dimensional spaces are dominant but more general cases must also be supported. Geographical Information Systems (GIS) as a Web service would support both educational curricula on the environment as well as the latest simulations of a new model for earthquake triggering. The general authoring web service of table 3 would need special extensions for science – in particular to support mathematical notation as seen in most leading word processing systems today. The network server model of chapters 24 and 25 (NetSolve and Ninf) is particularly appropriate for some generic science servers. The NEOS optimization resource at Argonne is a nice example of this type of service [13]. This of course developed a long time before Web services and illustrates that Web services are in many cases just following existing best practice. We illustrated the role of sensors in section 2 and "tenure evaluation" is listed to illustrate how general application Web services (in this case human resource service of table 3) are specialized in particular domains.

Table 5: Science and Engineering Research (e-Science)

Portal Shell Services	Job control/submission, scheduling, visualization, parameter specification, monitoring
Software Development Support	Wrapping, application Integration, version control, software engineering
Scientific Data Services	High Performance, special formats, virtual data
(Theory) Research Support Services	Scientific notebook/whiteboard, brainstorming, theorem proving
Experiment Support	Virtual Control Rooms (accelerator to satellite), Data analysis, virtual instruments, sensors (Satellites to field work to wireless to video to medical instruments, multi-instrument federation
Publication	Submission, preservation, review, uses general copyright service
Dissemination and Outreach	Virtual Seminars, Multi-cultural customization, multi-level presentations,

Table 5 illustrates some of the Web services that are needed by e-science. We have the suite of computational tools with a portal (controlling user-facing ports) front end described in section C of the book, chapters 20 to 34. Unlike education (table 6), we often require the highest performance both in simulation and communication services. Virtual data described in chapter 16 was developed to support research efforts with multiple data sources and multiple analysis efforts spread around the world – see chapters 38 and 39. This concept will also be important in distance education where one builds curricula and mentoring models with geographically distributed resources. For example, a student might take classes with grades from different organizations but these may need to be integrated to form learning plans. The near term impact of the Grid will perhaps be greater in experimental and phenomenological fields (due to the data deluge of chapter 36) than theoretical studies. Some of the needed experimental Web services are covered in part D of the book, chapters 35 to 43. However support for theoreticians interacting at a distance has similarities to those needed for education. We mentioned the latter needed excellent quality of service. The same is even more true for "e-theory" as the latter must support unstructured interactions at any time – the structure (known schedules) in educational class delivery (and office hours) helps one improve quality of service with careful pre-lesson testing [12]. Publication services are an interesting area with some possibilities enabled by the Grid discussed in chapter 36.

Dissemination and outreach for e-Science has close relations to the requirements of an education grid. A speaker can give their seminar from afar using similar technologies to those needed by distance education. The Web service model has a general and possibility important implication for the ongoing search for better ways to integrate research and education. Thus in e-science, we are instructed to build each electronic science application as a network of interlocking interacting Web services. Typically a "leading edge" research "Web service" (say a data analysis or simulation) cannot be easily used in education directly because its complexity often hides the "essential features" and because it needs substantial resources. Deriving an educational version of a research tour de force is both time consuming and not easy to "automate". The modularity of Web services offers an approach to this. In some cases, we can take a linked set of research Web services and "just" modify a few of them to get an educational version. This requires thinking carefully through the Web service implementation to isolate complexity (which needs to be simplified for education or outreach) in just a few places.

As already mentioned IMS and ADL have already defined many of the XML properties needed for education. This is illustrated by the list of available IMS specifications as of August 2002. These are [6]:

- Accessibility (universal access)
- Competency Definitions (grades, degrees, "learning outcomes", skills, knowledge)
- Content Packaging (defining collections such as courses built of lectures)
- Digital Repositories (should reflect directly digital libraries and content management)
- Enterprise (support people and groups such as organizations)
- Learner Information Package (education record, resume etc. of people)
- Meta-data including Dublin Core bibliographical information [14]
- Question & Test (quizzes)

- Simple Sequencing (of lesson components)

These are supported in IMS compliant LMS (Learning management Systems) to provide the functionalities illustrated in table 6. Essentially all these services are available in familiar systems, which are for the education academic environment Blackboard [15], and WebCT [16]. However as we discussed, current education standards are not built with a service architecture and so an interoperable Web service Grid with components from different vendors is not possible – ADL [7] however has tested interoperability within the current model successfully. We suggest that an important next step for the education community is to discuss a service architecture and agree on the needed Web service and Grid interoperable standards. We suggest these should not be built in isolation but rather adopt the hierarchical model described here so that for instance learning content management is built on broader commodity standards; further perhaps aspects of quizzes should build on survey web services etc. An important complication of the Web service model is this linkage between the services of different systems with often the more specialized applications "ahead" of the generic case. We need to develop good ways to cope with this.

Table 6: Education as a Web service
(LMS or Learning Management System)

Registration	Extends generic human resources service
Student Performance	Grading including transcripts
Homework	Submission, answers; needs performance and security services
Quizzes	Set and take – extends voting/survey service
Curriculum (Content)	Authoring, prerequisites, completion requirements, standards, extend generic authoring and data management services to get LCMS (Learning Content Management Systems)
Assessment	Related to refereeing and reference (tenure) services
Course Scheduling	Related to generic event scheduling service in collaboration service
Learning Plans	Builds on Curriculum and Student performance services. Support building of "degrees" with requirements
Learning	Integrate curriculum, collaboration and Knowledge discovery services
Mentoring and Teaching	Office hours, (virtual) classrooms
Distance Education	Asynchronous and Synchronous, integrate curriculum, quiz etc. services with generic collaboration services

Collaboration is a general Grid service of great importance in education. We stress the service model because as described in [39], it is far clearer how to support collaboration for Web services than for general applications. The latter's state is defined by a complex mix of input information from files, user events and other programs. The state of a Web service is uniquely defined by its initial conditions and message based input information – we ignore the subtle effects of different hosting environments which give different results from the same message based information. Either the state defining or the user-facing port messages can be replicated (multicast) to give a collaborative Web service. There is no such simple strategy for a general application. Thus we see significant changes if programs like Microsoft Word are in fact restructured as a Web service.

Currently collaboration support falls into broad classes of products: instant messenger and other such tools from the major vendors such as AOL, Microsoft and Yahoo; audio-video conferencing systems such as the Access Grid and Polycom [17] [18]; largely asynchronous peer to peer systems such as Groove Networks and JXTA [19] [20]; synchronous shared applications for "web conferencing" and virtual seminars and lectures from Centra, Placeware, WebEx, Anabas, Interwise and the public domain VNC [21-26]. We can expect the capabilities of these systems to be "unbundled" and built as Web services. For example shared display is the most flexible shared application model and it is straightforward to build this as a Web service. Such a Web service would much more easily work with aggregation portals like Jetspeed [27] from Apache; it could link to the universal access Web service to customize the collaboration for different clients. The current clumsy integrations of collaboration systems with learning management systems (LMS) would be simplified as we just need to know the LMS is a Web service and capture its input or output messages. We could hope that instant messengers would be integrated as another portlet in such a system; currently they come from different vendors and can only be easily linked to a distance education session using intermediaries like that from Jabber [28].

We have suggested that education is an important focus area for the Grid. The Grid offers a new framework which can exploit the sophisticated existing Object API's from IMS [6] and ADL [7] to build a Web service environment that can better enable e-education, which offers learners a much richer environment than available today. We expect education's rich XML structure to lead development of tools for handling distributed meta-data of complex structure. We can also expect education to be a natural application for peer-to-peer Grids.

References

1) IBM, Microsoft and BEA, *Business Process Execution Language for Web Services*, or BPEL4WS http://www-3.ibm.com/software/solutions/webservices/pr20020809.html Aug 9 2002.
2) Universal Description, Discovery and Integration (UDDI) project http://www.uddi.org/
3) Castor open source data binding framework for Java http://castor.exolab.org/
4) Web Services Description Language http://www.w3.org/TR/wsdl
5) Simple Object Access Protocol (SOAP) http://www.w3.org/TR/SOAP/
6) Instructional Management Systems (IMS). http://www.imsproject.org
7) Advanced Distributed Learning Initiative (ADL) http://www.adlnet.org
8) Fox G.C.," From Computational Science to Internetics: Integration of Science with Computer Science", in Ronald F. Boisvert and Elias Houstis (eds.), *Computational Science, Mathematics and Software*, Purdue University Press, West Lafayette, Indiana, , ISBN 1-55753-250-8. Pages 217-236, July 2002. http://grids.ucs.indiana.edu/ptliupages/publications/Internetics2.pdf
9) Geoffrey Fox, Experience with Distance Education 1998-2002, http://grids.ucs.indiana.edu/ptliupages/publications/disted/

10) Biology Workbench at SDSC (San Diego Supercomputer Center) http://workbench.sdsc.edu/
11) NSF PACI (Partnership in Advanced Computing Infrastructure) EOT (Education and Outreach) Program http://www.eot.org
12) David E. Bernholdt, Geoffrey C. Fox, Nancy J. McCracken, Roman Markowski, and Marek Podgorny, Reflections on Three Years of Network-Based Distance Education, unpublished report for US Army Corp of Engineers ERDC Vicksburg Miss, July 2000, http://grids.ucs.indiana.edu/ptliupages/publications/disted/erdctraining00.pdf
13) NEOS Optimization Server from Argonne National Laboratory http://www-neos.mcs.anl.gov/neos/
14) The Dublin Core bibliographic Meta data. http://dublincore.org/
15) Blackboard Learning System http://www.blackboard.com/
16) WebCT Learning system http://www.webct.com/
17) Access Grid Conferencing Environment from Argonne National Laboratory, http://www.accessgrid.org
18) Polycom Conferencing Environment http://www.polycom.com
19) Groove Desktop Collaboration Software, http://www.groove.net/
20) JXTA peer to peer environment from Sun Microsystems http://www.jxta.org
21) Centra Collaboration Environment. http://www.centra.com
22) Placeware Collaboration Environment. http://www.placeware.com
23) WebEx Collaboration Environment. http://www.webex.com
24) Anabas Collaboration Environment. http://www.anabas.com
25) Interwise Enterprise Communications Platform, http://www.interwise.com
26) Virtual Network Computing System (VNC). http://www.uk.research.att.com/vnc
27) Jetspeed Enterprise portal from Apache http://jakarta.apache.org/jetspeed/
28) Jabber Instant Messenger http://www.jabber.org/
29) Fran Berman, Geoffrey Fox and Tony Hey, "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons Ltd, Chichester, 2003.
30) Bill Johnston, "Implementing Production Grids", Chapter 5 in Ref. [29].
31) Ian Foster, Carl Kesselman and Steven Tuecke, "Anatomy of the Grid", Chapter 6 in Ref. [29].
32) Dennis Gannon, Rachana Ananthakrishnan, Sriram Krishnan, Madhusudhan Govindaraju, Lavanya Ramakrishnan and Aleksander Slominski, "Grid Web Services and Application Factories", Chapter 9 in Ref. [29].
33) Douglas Thain, Todd Tannenbaum, and Miron Livny, "Condor and the Grid", Chapter 11 in Ref. [29].
34) Pratap Pattnaik, Kattamuri Ekanadham and Joefon Jann, "Autonomic Computing and the Grid", Chapter 13 in Ref. [29].
35) Paul Watson, "Databases and the Grid", Chapter 14 in Ref. [29].
36) Peter Z. Kunszt and Leanne P. Guy, "The Open Grid Service Architecture and Data Grids", Chapter 15 in Ref. [29].
37) Reagan Moore and Chaitan Baru , "Virtualization Services for Data Grids", Chapter 16 in Ref. [29].
38) David De Roure, Nicholas Jennings and Nigel Shadbolt, "The Semantic Grid: A Future e-Science Infrastructure", Chapter 17 in Ref. [29].

39) Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko, Sangmi Lee, Shrideep Pallickara, Marlon Pierce, Xiaohong Qiu, Xi Rao Ahmet Uyar, Minjun Wang and Wenjun Wu, "Peer-to-Peer Grids", Chapter 18 in Ref. [29].
40) Wolfgang Hoschek, "Peer-to-Peer Grid Databases for Web Service Discovery", Chapter 19 in Ref. [29].
41) Geoffrey Fox, Dennis Gannon and Mary Thomas, "Overview of Grid Computing Environments", Chapter 20 in Ref. [29].
42) Geoffrey Fox and Shrideep Pallickara, "NaradaBrokering: An Event Based Infrastructure for Building Scaleable Durable Peer-to-Peer Grids", Chapter 22 in Ref. [29].