Emerging Grid Standards

Individual projects carried out to meet specific needs must interact as part of a larger Grid environment, but no international consensus exists as to which of the many ideas, proposed standards, and specifications are likely to dominate in the future.

Mark Baker
University of Portsmouth

Amy Apon
University of Arkansas

Clayton Ferner
Jeff Brown
University of North Carolina at Wilmington

The Grid can be seen as a framework for "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources."[1] It allows researchers in different administrative domains to use multiple resources for problem solving and provides an infrastructure for developing larger and more complex applications potentially faster than with existing systems.

In general terms, the Grid has evolved from a carefully configured infrastructure, which supported limited Grand Challenge applications executing on high-performance hardware among numerous US centers,[2] to what we are starting to see today—a seamless and dynamic virtual environment being driven by international development and take-up.

As the Grid's potential started to become a reality over the past few years, industry has become increasingly involved. Commercial participation has accelerated development of hardened, industrial-strength software that supports Grid environments outside academic laboratories. This in turn has impacted both the Grid's architecture and the associate protocols and standards.

Most profoundly, the recent adoption of Web services, while bringing significant benefits, has also produced a somewhat fragmented landscape for application developers. Software and Grid services developers ideally seek to conform to conventions and standards widely adopted by their community. However, for various political and technical reasons, there are now competing views of how to implement the architecture and what standards to follow. This infighting is inhibiting Grid developers, who lack the assurance that future standards will support those used today.

GRID-RELATED STANDARDS BODIES

The Global Grid Forum (www.ggf.org) is the primary standards-setting body for the Grid. The GGF works with many organizations throughout industry that influence Grid standards and policies, including those for security and virtual organizations.

Other bodies include the Organization for the Advancement of Structured Information Standards, the World Wide Web Consortium, the Distributed Management Task Force, the Web Services Interoperability Organization, groups within Internet2 such as the Peer-to-Peer Working Group and the Middleware Architecture Committee for Education, and the Liberty Alliance.

Global Grid Forum

The GGF is a community-driven set of working groups that are developing standards and best practices for wide-area distributed computing. It was formed in 1998 from the merger of the Grid Forum in North America, the Asia-Pacific Grid community, and the European Grid Forum (eGrid).

In a process similar to that used for Internet standards, the GGF creates four types of documents that provide information to the Grid community:

- *informational*—a useful idea or set of ideas;
- *experimental*—useful experiments;
- *community practice*—common practices or processes that influence the community; and
- *recommendations*—specifications, which are analogous to Internet standards-track documents.

The GGF currently divides its efforts among seven areas—including, for example, architecture, data,

0018-9162/05/$20.00 © 2005 IEEE

OGSA defines the fundamental services that an e-business or e-science application would use.

and security—within which numerous working and research groups operate. Within the data area, standards under development include data access and integration services, Grid file systems, Grid FTP, grid storage, IPv6, and data replication. Nearly 30 research groups explore longer-term issues for which it may be premature to develop specifications.

Joining a GGF working group involves simply subscribing to its e-mail list. The project members, meeting agendas, and work progress are all posted online.

OASIS

A not-for-profit international organization that promotes industry standards for e-business, OASIS (www.oasis-open.org) was founded in 1993 as SGML Open and changed its name in 1998 to reflect its expanded technical scope. This includes developing standards such as those related to the Extensible Markup Language (XML) and the universal description, discovery, and integration (UDDI) service. OASIS produces Web services standards that focus primarily on higher-level functionality such as security, authentication, registries, business process execution, and reliable messaging.

Participants in OASIS can be either unaffiliated individuals or member-company employees. At least three organizations must implement a standard before OASIS will approve it.

World Wide Web Consortium

The W3C (www.w3.org) is an international organization initiated in 1994 by Tim Berners-Lee to promote common and interoperable protocols. It created the first Web services specifications in 2003 and initially focused on low-level, core functionality such as SOAP and the Web Services Description Language (WSDL). The W3C has developed more than 80 technical specifications for the Web, including XML, HTML, and DOM.

W3C members are organizations that typically invest significant resources in Web technologies. OASIS is a member, and the W3C has partnered with the GGF in the Web services standards area.

Distributed Management Task Force

The DMTF (www.dmtf.org) is an industry-based organization founded in 1992 to develop management standards and integration technologies for enterprise and Internet environments. DMTF technologies include the Common Information Model and Web-Based Enterprise Management. The DMTF formed an alliance with the GGF in 2003[3] for the purpose of building a unified approach to the provisioning, sharing, and management of Grid resources and technologies.

Web Services Interoperability Organization

WS-I (www.ws-i.org) is an open industry body formed in 2002 to promote the adoption of Web services and interoperability among different Web services implementations. Its role is to integrate existing standards rather than create new specifications. WS-I publishes profiles that describe in detail which specifications a Web service should adhere to and offer guidance in their proper usage. The goal is to provide a set of rules for integrating different service implementations with a minimum number of features that impede compatibility.

Internet2

Internet2 (www.internet2.edu) is a consortium of groups from academia, industry, and government formed in 1996 to develop and deploy advanced network applications and technologies.

The Middleware Architecture Committee for Education (http://middleware.internet2.edu/MACE) aims to create an interoperable middleware infrastructure for research and education. MACE develops good-practices documents, designs pilot projects and intercampus experiments, and recommends technical standards. Internet2 working groups related to Grid standards include the Higher Education PKI Technical Activities Group, the Peer-to-Peer Working Group, and the Shibboleth project.

Liberty Alliance

The Liberty Alliance (www.projectliberty.org) is an international alliance of companies, nonprofit groups, and government organizations formed in 2001 to develop an open standard for *federated identity management,* which addresses technical, business, and policy challenges surrounding identity and Web services. The Liberty Alliance has developed the Identity Federation Framework, which enables identity federation and management and provides interface specifications for personal identity profiles, calendar services, wallet services, and other specific identity services.

OPEN GRID SERVICES ARCHITECTURE

The most important Grid standard to emerge recently is the Open Grid Services Architecture, which aims to define a common, standard, and open architecture for Grid-based applications.

The GGF announced OGSA at Global Grid Forum 4 in February 2002, presented a draft overview four months later,[4] and created the OGSA Working Group in September 2002 to draft specifications.

At GGF10 in March 2004, the GGF declared OGSA to be its flagship architecture, and three months later, at GGF11, it released version 1.0.[5] OGSA v2.0, a proposed GGF recommendation, is scheduled for release in June 2005.

Service-oriented architecture

As Figure 1 shows, OGSA is a service-oriented architecture that specifies a set of distributed computing patterns realized using Web services. It aims to define all the fundamental services that an e-business or e-science application would use such as job and resource management, communications, and security, leaving various working groups within the GGF and other Grid-standards organizations to specify the services' interfaces, semantics, protocols, and other technical details.

Because the Grid is a dynamic environment in which service instances can come and go during task dispatching, resource configuration and provisioning, and system state changes, OGSA provides interfaces for lifecycle service management. It also supports state data associated with Grid services, an approach conceptually similar to traditional object-oriented programming environments. In addition, OGSA includes a callback operation in which clients can register interest in a service and receive notification of any change in that service.

Open Grid Services Infrastructure

OGSA instantiations depend on emerging specifications. The first instantiation was the Open Grid Services Infrastructure. OGSI was based on the concept of *Grid services,* enhanced Web services that provided a standard set of mechanisms to manage state. Released in July 2003, OGSI v1.0 defined a set of principles and extensions for using WSDL and XML Schema to enable stateful Web services.[6]

Critics identified several problems with OGSI.[7] First, many thought it was too large for one specification. In addition, because OGSI was not a pure subset of Web services, it required a modification to standard WSDL, called Grid WSDL, which would have necessitated extending current tools to parse and process WSDL for Grid services. Finally, even though many other Web services systems have object-oriented implementations, some viewed OGSI as too object oriented. To support transient, potentially short-lived instances, OGSI used OO concepts such as statefulness and the factory pattern to create Grid service instances.

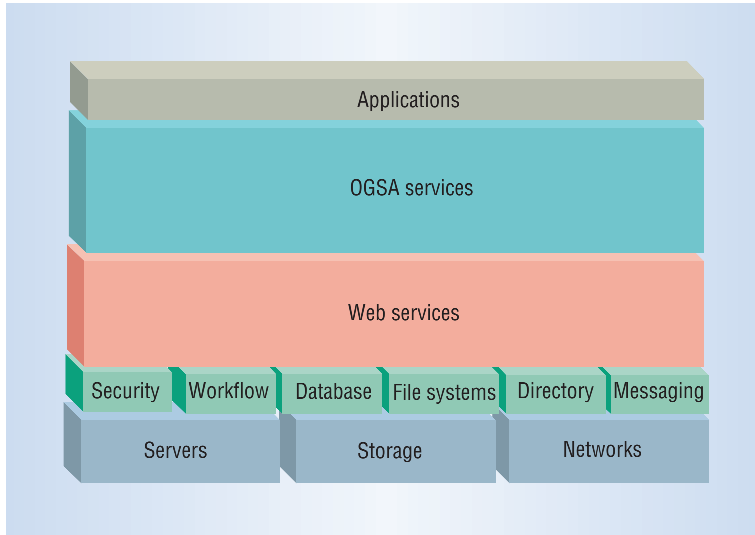

Figure 1. Open Grid Services Architecture. OGSA is a service-oriented architecture that specifies a set of distributed computing patterns realized using Web services.

WEB SERVICES RESOURCE FRAMEWORK

Widespread dissatisfaction with OGSI led to a collaborative effort among architects from the Grid and Web services communities to define an alternative infrastructure based on unadulterated Web services specifications. On 20 January 2004, Hewlett-Packard, IBM, Fujitsu, and the Globus Alliance announced the WS-Resource Framework (www.globus.org/wsrf). WSRF contains a set of specifications for expressing the relationship between stateful resources and Web services. The specifications define specific message exchange formats and related XML definitions.

After revising and updating the WSRF specifications based on industry feedback, a development team submitted the final results to two new OASIS technical committees, the WS-Resource Framework (WSRF) TC and the WS-Notification (WSN) TC.

The WSRF TC (www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf) was formed to standardize four specifications:

- WS-ResourceLifetime—describes how to manage the lifetime of a resource and specifies Web services operations used to destroy a WS-Resource;
- WS-ResourceProperties—defines how to query and modify WS-Resources described by XML Resource Property documents;
- WS-ServiceGroup—describes how to represent and manage collections of Web services and/or WS-Resources; and
- WS-BaseFaults—defines a base fault XML type for use when returning faults in a Web services message exchange.

WSRF defines functions that allow interaction with WS-Resources such as query, lifetime management, and group membership.

The WSN TC (www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn) was created to standardize three other specifications defining Web services interfaces:

- WS-BaseNotification—handles asynchronous notification, including interfaces used by a notification producer or consumer;
- WS-BrokeredNotification—handles asynchronous notification; and
- WS-Topics—organizes and categorizes items of interest for subscription, known as topics.

Both technical committees republished the specifications as working drafts and started reviewing them in depth.

Stateful resources

The OASIS WSRF TC aims to define a generic and open framework for modeling and accessing stateful resources using Web services. This includes mechanisms to describe views of state, support state management through properties associated with the Web service, and describe how these mechanisms are extensible to groups of Web services.

WSRF defines the means by which

- a Web service can be associated with one or more stateful resources;
- a service requestor can access stateful resources indirectly through Web services that encapsulate the state and manage all aspects of the service-based access to the state;
- the stateful resources can be destroyed, immediately or via time-based destruction;
- a stateful resource's type definition can be associated with a Web service's interface description and ensure well-formed queries against the resource via its interface;
- a stateful resource's actual state can be queried and modified via message exchanges;
- end-point references to a Web service that encapsulate stateful resources can be renewed when they become invalid due to, for example, a transient failure in the network; and
- the stateful resources can be aggregated for domain-specific purposes.

At the heart of WSRF is WS-Resource, which defines the relationship between Web services and stateful resources as an *implied resource pattern*. A WS-Resource is the "composition of a Web service and a stateful resource"[7] that can be described by an XML Schema associated with the Web services port type and addressed by a WS-Addressing EndpointReference.[8] WSRF defines functions that allow interaction with WS-Resources such as query, lifetime management, and group membership.

Currently, several early releases of WSRF-based systems are available, including Globus Toolkit 4 (www-unix.globus.org/toolkit) and WSRF.NET (www.cs.virginia.edu/~gsw2c/wsrf.net.html). Other development teams have implementations in progress such as WSRF::Lite (www.omii.ac.uk/mp/mp_wsrf_lite.htm), Unicore (www.unicore.org), and Python Globus (http://dsd.lbl.gov/gtg/projects/pyGlobus).

Event notification

Currently, two specifications describe event notification with respect to resources: WS-Eventing and WS-Notification. Originally released in January 2004, WS-Eventing[9] is a collaborative effort by Microsoft, IBM, BEA Systems, Computer Associates International, Sun Microsystems, and Tibco Software. Released around the same time, WS-Notification (www-106.ibm.com/developerworks/library/specification/ws-notification) is a joint initiative by Akamai Technologies, Computer Associates International, Fujitsu Laboratories of Europe, Globus, Hewlett-Packard, IBM, SAP AG, Sonic Software, and Tibco Software.

There is a move to merge these competing specifications, especially as IBM, Computer Associates, and Tibco contribute to both. The OASIS WSN TC is currently developing a standard based on WS-Notification.

WS-Eventing. This specification allows Web services to be notified of events that occur with other services. An *event source* is a Web service that produces notifications or event messages. An *event sink* is a Web service that receives notifications. A Web service subscribes itself or another service with a source to be a sink and thus receive events from that source. The subscription has an expiration time, which can be renewed, although it may have an indefinite termination.

WS-Eventing defines a *subscription manager,* which manages the subscriptions on behalf of an event source. It also includes the concept of *delivery mode,* which specifies how notifications should be delivered. For example, a source service can request that a notification be wrapped in a standard message. The only mode that the specification defines is *push mode,* which implies the delivery of individual, unsolicited, asynchronous SOAP messages. WS-Eventing also provides for source-side filtering of messages, such as using an XPath predicate expression.

WS-Notification. This family of specifications describes the mechanisms by which Web services can receive notification of an event related to a resource.[10] Web services that produce notifications are referred to as *notification producers,* while those that receive such notifications are *notification consumers*.

WS-Notification also describes a *subscription manager* as well as a *notification broker.* Using a separate notification broker can

- relieve the producer of the load needed to process notifications;
- reduce the number of interserver messages;
- provide a finder service, matching producers and consumers; and
- allow anonymous notification.

The notification producer can perform both of these roles, or a separate entity can offload these responsibilities from the producer. Unlike the subscription manager in WS-Eventing, this role in WS-Notification provides mechanisms to pause and resume subscriptions as well as to list them. Both specifications enable a separate entity to make subscription requests on behalf of a notification consumer.

Topics in WS-Notification support the hierarchical organization of notifications and offer a convenient way to locate notifications of interest. It is not clear whether topics provide greater functionality than XPath with respect to filtering XML documents, but topics should be applicable to other types of documents. Further, using topics in combination with the notification broker to pause and resume subscriptions enables demand-based publishing: If there are no subscribers, then nothing is published.

OTHER STANDARDS AND TRENDS

Despite the upcoming release of OGSA v2.0, some ongoing and recently initiated Grid projects cannot wait for production implementations of WSRF. Alternatives include WS-I's Basic Profile 1.0, the Web Services Grid Application Framework, and the Open Middleware Infrastructure Institute's WS-I+.

WS-I Basic Profile

In April 2004, WS-I published Basic Profile 1.0,[11] which contains guidelines for using SOAP, WSDL, and UDDI. BP1.0 has both recommendations and requirements for compliant services—for example, it recommends sending SOAP messages with HTTP/1.1 but requires the use of either HTTP/1.1 or HTTP/1.0.

Many applications other than Web services use HTTP, which has features that are appropriate in some environments but not in others. For example, HTTP cookies facilitate Web-based state management, but because cookies are not part of the SOAP envelope, BP1.0 mandates their use only in limited ways.

In some cases, BP1.0 tightens requirements in existing specifications. For example, SOAP 1.1 allows the use of the HTTP POST method as well as the HTTP Extension Framework's M-POST method, whereas BP1.0 permits only the former.

BP1.0 also clarifies ambiguities in some specifications. For example, a service sends a SOAP fault message when an error occurs. BP1.0 requires that the `soap:fault` element has no element children other than `faultcode`, `faultstring`, `faultactor`, and `detail`. Further, for extensibility the detail element can contain any type of element, thus a compliant service must accept such messages.

WS-I released Basic Profile 1.1[12] in August 2004. Some of the material in BP1.0 became Simple SOAP Binding Profile 1.0.[13] WS-I also released Attachments Profile 1.0[14] in August 2004.

WS-Notification describes the mechanisms by which Web services can receive notification of an event related to a resource.

Web Services Grid Application Framework

Grid services have requirements beyond those of standard Web services. The Web Services Grid Application Framework[15] proposes to meet the needs of Grid applications by extending basic Web services functionality.

The WS-GAF approach differs greatly from OGSI. Consider, for example, the problem of making services stateful. With OGSI, the user creates a service instance that generally only the creator uses. In contrast, WS-GAF uses the WS-Context specification,[16] which mandates that SOAP message headers carry service context information.

WS-I+

The UK e-Science Programme (www.rcuk.ac.uk/escience) has funded more than 100 separate projects that use a number of Grid technologies, many of which are based on Web services. It has also established the Open Middleware Infrastructure Institute (www.omii.ac.uk) to act as a center for expertise in Grid middleware and a repository for the software developed by the various projects. One goal of the OMII is to provide a relatively stable

The OMII expects to exploit BPEL's built-in extensibility mechanisms to support the scientific community's Web services needs.

development environment for Grid-based enterprises.

The lack of Grid standards is a serious problem for the e-Science projects, some of which will be complete before specifications such as WSRF and WSN emerge. The OMII's approach is to build on WS-I profiles and create WS-I+,[17] which will identify existing standards that are considered safe and will potentially interoperate with emerging specifications. As in WS-I, the core of the service architecture consists of XML Schema Definition, WSDL 1.1, and SOAP 1.1.

For service discovery, the WS-I profiles include UDDI; WS-I+ might use UDDI, although the OMII is considering adopting registry service extensions that better suit scientific application needs. To address Grid workflow, WS-I+ uses the popular Business Process Execution Language (www-128.ibm.com/developerworks/library/specification/ws-bpel). The OMII expects to exploit BPEL's built-in extensibility mechanisms to support the scientific community's Web services needs.

Two competing specifications deal with addressing Grid services: WS-Addressing[8] and WS-MessageDelivery.[18] WS-Addressing has not been submitted to a standards body but is part of WSRF, while WS-MessageDelivery has been submitted to the W3C. WS-I+ will include WS-Addressing, which should facilitate future integration with WSRF.[19]

GRID SECURITY INFRASTRUCTURE

The Grid Security Infrastructure (https://forge.gridforum.org/projects/gsi-wg) implemented by the Globus Toolkit is a de facto standard for Grid security. GSI uses X.509 identity and proxy certificates, which provide a globally unique identifier that can authenticate and authorize an entity with accessed Grid resources.[20] In GSI, the owner typically grants use of a resource to individual users, who must have an account for each accessed resource. This becomes impractical as the number of users and resources grows.

Community Authorization Service

To overcome the access problem, the Community Authorization Service[21] provides an individual community identifier that authorizes a user for a resource. However, this solution requires additional Grid infrastructure and administration, which can lead to security problems when unknown users request a CAS account. For example, the CAS administrator might not know the person's institutional affiliation, which can be used to verify identity and trustworthiness.

GridShib and ESP-GRID

Two new projects are investigating alternative solutions that will impact the GSI standards. GridShib (http://grid.ncsa.uiuc.edu/GridShib) and ESP-GRID (http://e-science.ox.ac.uk/oesc/projects/index.xml.ID=body.1_div.20) will create new mechanisms and policies for distributed authorization and help Grid virtual organizations integrate with traditional organizations' security infrastructures. These projects should also lead to new tools and standards for administering user attributes and resource requirements. Both projects will leverage technologies in the Internet2's Shibboleth project (http://shibboleth.internet2.edu).

Shibboleth. Based on the Security Assertion Markup Language standard (www.oasis-open.org/committees/tc_home.php?wg_abbrev=security), this system is designed to exchange attributes between trusted organizations to authenticate and authorize users to remote resources. A user who desires to access a resource at a remote institution authenticates at a home institution, then the home institution passes the user's attributes securely through a trust relationship to the remote institution.

The remote institution authorizes access to the resource based on the user's attributes. For example, a member of a biomedical informatics research group could receive access to a remote institution's data set based on this group membership. The remote institution can require any number of user attributes before granting access to the resource, and users have the option of releasing attributes to particular resources, thereby maintaining privacy for access to some types of remote resources. Shibboleth's approach simplifies access control policies and makes them more scalable.

GridShib. Funded by the National Science Foundation Middleware Initiative, GridShib supports an identity federation between the Grid and higher-education communities by combining Shibboleth with GSI. Currently, Shibboleth only provides authorization and authentication for Web-based resources. In addition to using existing campus authentication and identity management infrastructures, GridShib plans to provide access to non-Web-based resources.

To accomplish this, GridShib will introduce two new modes of operation. In *pull mode*, a user with a GSI certificate contacts Shibboleth with a registration request and sends a key certificate to the target resource; the target resource contacts Shibboleth with a request for the user's attributes based on the user's key certificate. In *push mode*, Shibboleth passes the user's attributes to the target resource along with the key certificate when the user requests access to the resource.

An initial release of GridShib, planned for summer 2005, will support the pull-mode operation; follow-on releases will support the push mode and user pseudonymity. GridShib will likely be integrated with Globus Toolkit 4.2 or 4.4 and Shibboleth v1.3.

ESP-GRID. Funded by the UK's Joint Information Systems Committee, ESP-GRID is also investigating how Shibboleth can help provide solutions to Grid authentication, authorization, and security issues. In addition, ESP-GRID will reappraise public-key infrastructure use within the UK e-Science Programme and the Grid in general.

Any developer who wants to create Grid services or applications today faces the dilemma of deciding which of the many frameworks and specifications to follow, as currently there are no guarantees that industry and the open source community will embrace any one of them. Research is one thing, wide-scale deployment another.

Developers could use any of the competing frameworks and specifications to build wide-area infrastructure and associated applications. However, apart from the core Web services components—SOAP 1.2 and WSDL 1.1—all of the specifications are relatively new. In addition, many are drafts or in an early definition stage; even if a particular specification is accepted, a process that can take several years, its exact form is likely to differ from earlier versions. For these reasons, Savas Parastatidis and Jim Webber[22] argue that for production services the safest approach is to adopt existing and stable Web services specifications.

Despite resistance to new specifications, there is a growing demand for standards at the Grid's higher-level layers. For example, a recent effort among application developers to create a job and file management standard is gaining momentum. In December 2004, the GGF established a research group to begin developing the Simple API for Grid Applications (https://forge.gridforum.org/projects/saga-rg), which an application developer could use to specify a job request along with associated files and resources. This API will replace a number of incompatible tools that developers currently are using for such tasks. The SAGA research group will probably become a full GGF working group in the near future.

Another needed higher-level Grid standard would specify the type of information that a Grid monitoring system gathers. Although many open source and commercial tools are available for this purpose, the data returned varies considerably in content and detail. Lack of a standard makes monitoring heterogeneous resources difficult and limits the ability to assign tasks to resources and perform adaptive metascheduling.

OGSA and WSRF represent significant cooperation among researchers in academia, government, and industry. These joint efforts point to a promising future for the Grid regardless of the uncertainties, inconsistencies, and interoperability problems developers currently face. ■

For production services, the safest approach is to adopt existing and stable Web services specifications.

References

1. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputer Applications,* vol. 15, no. 3, 2001, pp. 200-222.
2. D. De Roure et al., "The Evolution of the Grid," *Grid Computing: Making the Global Infrastructure a Reality,* F. Berman, G. Fox, and A.J.G. Hey, eds., John Wiley & Sons, 2003, pp. 65-100.
3. "DMTF and GGF Announce Alliance Partnership," *Grid Today,* 5 May 2003; www.gridtoday.com/03/0505/101375.html.
4. I. Foster et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," draft document, 22 June 2002; www.globus.org/research/papers/ogsa.pdf.
5. I. Foster et al., *The Open Grid Services Architecture,* v1.0, Global Grid Forum, GWD-I (draft-ggf-ogsa-spec-019), 12 July 2004; www.ggf.org/documents/Drafts/draft-ggf-ogsa-spec.pdf.
6. S. Tuecke et al., *Open Grid Services Infrastructure (OGSI),* v1.0, Global Grid Forum, GFD-R-P.15 (proposed recommendation), 27 June 2003; www.ggf.org/documents/GWD-R/GFD-R.015.pdf.
7. K. Czajkowski et al., "From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution," v1.1, *IBM DeveloperWorks*, 5 Mar. 2004; /www-106.ibm.com/developerworks/library/ws-resource/ogsi_to_wsrf_1.0.pdf.
8. D. Box et al., *Web Services Addressing (WS-Addressing),* W3C member submission, 10 Aug. 2004; www.w3.org/Submission/2004/SUBM-ws-addressing-20040810.

9. D. Box et al., *Web Services Eventing (WS-Eventing)*, Aug. 2004; http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-eventing.asp.
10. S. Graham et al., *Web Services Notification (WS-Notification)*, v1.0, 20 Jan. 2004; http://ifr.sap.com/ws-notification/ws-notification.pdf.
11. K. Ballinger et al., *Basic Profile*, v1.0, Wed Services Interoperability Organization, 16 Apr. 2004; www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html.
12. K. Ballinger et al., *Basic Profile*, v1.1, Web Services Interoperability Organization, 24 Aug. 2004; www. ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html.
13. M. Nottingham, *Simple SOAP Binding Profile*, v1.0, Web Services Interoperability Organization, 24 Aug. 2004; www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html.
14. C. Ferris, A. Karmarkar, and C.K. Liu, *Attachments Profile*, v1.0, Web Services Interoperability Organization, 24 Aug. 2004; www.ws-i.org/Profiles/AttachmentsProfile-1.0-2004-08-24.html.
15. S. Parastatidis et al., *A Grid Application Framework Based on Web Services Specifications and Standards,* North East Regional e-Science Centre, School of Computing Science, Univ. of Newcastle upon Tyne, UK, 2003; www.neresc.ac.uk/ws-gaf/A%20Grid%20Application%20Framework%20based%20on%20Web%20Services%20Specifications%20and%20Practices%20v1.0.pdf.
16. M. Little, E. Newcomer, and G. Pavlik, *Web Services Context Specification (WS-Context)*, draft version 0.8, 3 Nov. 2004; http://xml.coverpages.org/WS-ContextCD-9904.pdf.
17. M. Atkinson et al., "Web Service Grids: An Evolutionary Approach," Oct. 2004; www.omii.ac.uk/paper_web_service_grids.pdf.
18. A. Karmarkar et al., *WS-MessageDelivery* v1.0, W3C member submission, 26 Apr. 2004; www.w3.org/Submission/2004/SUBM-ws-messagedelivery-20040426.
19. N. Leavitt, "Are Web Services Finally Ready to Deliver?" *Computer,* Nov. 2004, pp. 14-18.
20. V. Welch et al., "X.509 Proxy Certificates for Dynamic Delegation," *Proc. 3rd Ann. PKI R&D Workshop,* NIST, 2004; www.globus.org/Security/papers/pki04-welch-proxy-cert-final.pdf.
21. L. Pearlman et al., "A Community Authorization Service for Group Collaboration," *Proc. IEEE 3rd Int'l Workshop Policies for Distributed Systems and Networks*, IEEE CS Press, 2002, pp. 50-59; www.globus.org/research/papers/CAS_2002_Revised.pdf.
22. S. Parastatidis and J. Webber, "Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications," *Proc. 2004 IEEE Int'l Conf. Services Computing,* IEEE CS Press, 2004, pp. 65-72.

***Mark Baker** is a reader in distributed systems and heads the Distributed Systems Group at the University of Portsmouth, UK. His research interests include wide-area resource monitoring, the integration of information services, and cross-environment application and service development. Baker received a PhD in maritime technology from Cardiff University, Wales. He is a member of the IEEE and the IEEE Computer Society, and is also cofounder and cochair of the IEEE Computer Society's Technical Committee on Scalable Computing. Contact him at mark.baker@computer.org.*

***Amy Apon** is an associate professor in the Department of Computer Science and Computer Engineering at the University of Arkansas. Her research interests include cluster computing, security and privacy in Grid systems, and distributed file systems. Apon received a PhD in computer science from Vanderbilt University. She is a member of the IEEE, the IEEE Computer Society, and the ACM. Her research is supported by an Extending the Reach grant from Educause and grant number DUE-0410966 from the National Science Foundation. Contact her at aapon@uark.edu.*

***Clayton Ferner** is an assistant professor in the Department of Computer Science at the University of North Carolina at Wilmington. His research interests include Grid computing and parallel and distributed computing. Ferner received a PhD in mathematics and computer science from the University of Denver. He is a member of the IEEE, the IEEE Computer Society, and the ACM. Contact him at cferner@uncw.edu.*

***Jeff Brown** is a professor in the Department of Mathematics and Statistics at the University of North Carolina at Wilmington, where he also heads the software development team for the GridNexus Grid computing project. His research interests include computer-aided geometric design and computational geometry. Brown received a PhD in mathematics from the University of Georgia. Contact him at brownj@uncw.edu.*