Semantics and Knowledge Grids: Building the Next-Generation Grid

Mario Cannataro, *University Magna Græcia of Catanzaro*

Domenico Talia, *University of Calabria*

Evolving grid resources, tools, and applications offer a means of dealing intelligently with enormous quantities of data. The authors survey the field and propose a software architecture that integrates semantic modeling and knowledge discovery with grid technologies.

Just as the Internet is shifting its focus from information and communication to a knowledge delivery infrastructure, we see the Grid moving from computation and data management to a pervasive, worldwide knowledge management infrastructure. Driving the Grid's further evolution is the necessity of dealing with the enormous amount of data produced at a rate never seen before. So-called data tombs—data stores, where data is stored and often never accessed again—are appearing in many areas where data consumption is slower than production. Although we can imagine larger and more powerful databases and data warehouses in which to store data, humans or programs will access only a small portion of it. We have the technology to store and access data, but we seem to lack the ability to transform data tombs into useful data and extract knowledge from them.[1]

Here, we review some of the current and future technologies that will impact the architecture, computational model, and applications of future grids. We attempt to forecast the evolution of computational grids into what we call the *next-generation grid*, with a particular focus on the use of semantics and knowledge discovery techniques and services. We propose a comprehensive software architecture for the next-generation grid, which integrates currently available services and components in Semantic Web, Semantic Grid, P2P, and ubiquitous systems. We'll also discuss a case study that shows how some new technologies can improve grid applications.

Emerging technologies

Current and future technologies will play a major role in the development of future grids. (See the sidebar for background information on how grids developed.) Among them, we can identify those related to high-level grid use and grid applications, such as the Semantic Grid and knowledge discovery services, and others that relate to architecture issues, such as the Open Grid Services Architecture (OGSA) and peer-to-peer (P2P) computing models.

The Semantic Web

The Semantic Web is defined as "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."[2] It aims to allow Web entities (software agents, users, and programs) to interoperate, dynamically discovering and using resources, extracting knowledge, and solving complex problems.

Although we need *metadata* (which annotates and describes Web content) to allow machine-to-machine operation, we need complex automated processing to give semantic meaning to each Web resource. A layered model of the Semantic Web comprises

- A set of Web resources, with a unique, global identity, described by metadata in a common and shared formalism, and with rules for inferring new metadata and knowledge through ontologies.
- A set of basic services, such as reasoning and querying over metadata, and ontologies and semantic search engines. These services represent a great improvement over current Internet services, such as the Domain Name System (DNS) and key-based search.
- A set of high-level applications developed by using basic services.

At this stage, major efforts are addressing the

1094-7167/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

Grids Past and Present

Since their birth, computational grids[1] have traversed different phases or generations. In the early 1990s, first-generation grids let us connect large supercomputing centers and aggregate computational power not available in the individual sites. They also let us decompose and coordinate distributed computations over thousands of private workstations. These early generation grid systems were the first real implementations of metacomputing and gave the basis for second-generation grids.[2]

Second-generation grids can link more than just a few regional or nationwide supercomputing centers. These grids use standards—such as HTTP, LDAP (lightweight directory access protocol), and PKI (public-key infrastructure)—that enable the deployment of a global-scale computing infrastructure, linking remote resources and allowing for collaboration between virtual organizations. From an architectural point of view, second-generation grids use grid middleware as glue between heterogeneous distributed systems, resources, users, and local policies. Grid middleware targets technical challenges in such areas as communication, scheduling, security, information, data access, and fault detection. Primary representatives of second-generation grids include systems such as Globus, UNICORE (*uni*form *i*nterface to *co*mputing *re*sources), and Legion.

Ian Foster and his colleagues delineated a milestone between second- and third-generation grids when they defined the Grid as a "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources—what we refer to as virtual organizations."[3] The motivation for third-generation grids was to simplify and structure the systematic building of grid applications through software component composition and reuse and the development of knowledge-based services and tools. Following the trend that has emerged in the Web community, service-oriented models have recently been proposed—for example, the Open Grid Services Architecture.[4,5]

References

1. I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann, 1999.
2. D. De Roure et al., "The Evolution of the Grid," *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, A.J.G. Hey, and G. Fox, eds., John Wiley & Sons, 2003, pp. 65–100.
3. 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. 6–13.
4. I. Foster et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Globus Project, 22 June 2002; www.globus.org/research/papers/ogsa.pdf.
5. D. Talia, "The Open Grid Services Architecture: Where the Grid Meets the Web," *IEEE Internet Computing*, vol. 6, no. 6, Nov./Dec. 2002, pp. 67–71.

development of languages and technologies for the standard modeling and implementation of metadata and ontologies such as XML Schema and RDF Schema, DAML+OIL, and OWL (www.daml.org, www.w3.org). Examples of tools and techniques, in their early stages, for manipulation and navigation include

- Ontology-building tools that let users define and build ontologies—for example, DUET (DAML UML enhanced tool), OilEd, Protégé and OntoEdit
- Ontology-based annotation tools, for annotating Web resources according to an ontology—for example, UBOT (UML-based ontology toolset) DAML
- Ontology-learning tools, for learning ontologies from natural language documents—for example, Corporum and Text-To-Onto
- Ontology manipulation tools—for example, DAML and Jena APIs—that let users navigate and manipulate ontologies

The Semantic Grid

The UK Engineering and Physical Sciences Research Council and Department of Trade and Industry's Core e-Science program started its Semantic Grid initiative, aiming to integrate and bridge the efforts made in the Grid and Semantic Web communities.[3] The Semantic Grid seeks to incorporate the Semantic Web approach into the ongoing Grid: "As the Semantic Web is to the Web, so is the Semantic Grid to the Grid. Rather than orthogonal activities, we see the emerging Semantic Web infrastructure as an infrastructure for grid computing applications." The Global Grid Forum supports the Semantic Grid through its Semantic Grid Research Group.

The Semantic Grid's research issues align with many aspects of the next-generation grid:

- Full support of a grid's three recognized layers: computation and data, information (where data produces information), and knowledge (where knowledge can be used to make decisions)
- Provision of seamless, pervasive, and secure resource use

Although the Semantic Grid initiative is still developing, we think it will be a significant component of the next-generation grid. Using semantics and ontologies in grids can offer high-level support for managing grid resources and for designing complex applications that will benefit from the use of semantics.

The Open Grid Services Architecture

OGSA introduces service orientation in grids, leveraging the results of Web services.[4] A *grid service* is a Web service that conforms to a set of conventions for controlled, fault-resilient, and secure management of stateful services and exposes capabilities via standard interfaces.[5] The OGSA model represents a border between second- and third-generation grids (see the "Grids Past and Present" sidebar).

To satisfy grid computing's new requirements, grid services significantly extend Web services as follows:

- Grid services can be dynamic and transient—that is, you can switch on or off some or all services (software, sensors, and computing resources) participating in computation, changing their availability.
- Grid services are globally distributed, without centralized control or a globally agreed trust relationship.
- Grid applications can involve tens to hundreds of grid services, which require efficient coordination.

The Knowledge Grid

Next-generation grids must be able to produce, use, and deploy knowledge as a basic element of advanced applications. In this scenario, we designed the Knowledge Grid system as a joint research project of ICAR-CNR (Istituto di Calcolo e Reti ad Alte Prestazioni-Consiglio Nazionale delle Ricerche) and the Universities of Calabria and Catanzaro, aiming at the development of an environment for geographically distributed high-performance knowledge discovery applications.[1] The Knowledge Grid is a high-level system for providing grid-based knowledge discovery services.[2] These services let professionals and scientists create and manage complex knowledge discovery applications composed as workflows that integrate data sets, mining tools, and computing and storage resources provided as distributed services on a grid.

Knowledge Grid facilities let users compose, store, share, and execute these knowledge discovery workflows and publish them as new components and services on the Grid. You can use the Knowledge Grid to mine very large data sets available over grids, make scientific discoveries, improve industrial processes and organization models, and uncover valuable business information. You can find other examples of knowledge grids elsewhere.[2]

The knowledge-building process in a distributed setting involves data and information collection, generation, and distribution followed by the collective interpretation of processed information into "knowledge." Knowledge building depends not only on data analysis and information processing but also on interpretation of the produced models and knowledge filtering. Knowledge discovery includes mechanisms for evaluating the correctness, accuracy, and usefulness of processed data sets, developing a shared understanding of the information, and filtering knowledge to be kept in an accessible organizational memory. The Knowledge Grid provides a higher level of abstraction and a set of services based on the use of grid resources to support all phases of the knowledge discovery process. So, it lets end users concentrate on developing the knowledge discovery process without worrying about grid infrastructure details.

The Knowledge Grid architecture is composed of a set of services divided into two layers (see Figure A):

- The Core K-Grid layer that interfaces basic and generic grid middleware and services
- The High-level K-Grid layer that interfaces the user by offering a set of services for the design and execution of knowledge discovery applications

The Knowledge Grid environment represents discovery processes as workflows that a user can compose using both concrete and abstract grid resources. Users define knowledge discovery workflows through a visual interface that shows resources (data, tools, and hosts) and offers mechanisms for integrating them in a workflow. The environment stores single resources and workflows using an XML-based notation that represents a workflow as a data flow graph of nodes, with each node representing either a data mining task or data transfer service. The XML representation allows workflows for discovery processes to be easily validated, shared, and translated into executable scripts and stored for future executions.

Recently, we implemented Vega (visual environment for grid applications), a software prototype that implements the main components of the Knowledge Grid environment, comprising services and functionalities ranging from information and discovery services to visual design and execution facilities. We can simplify the design and execution of complex applications by exploiting the advantages that a grid environment offers in the development of distributed knowledge discovery applications. The application design facility lets users build typical knowledge-based grid applications in an easy, guided, and con-

- Grid applications are often long-lived, impacting the lifetime requirement of grid services.

OGSA's Globus Toolkit 3 extends the previous version (GT2), providing open source implementation of the Open Grid Services Infrastructure. OGSI addresses detailed specifications of the interfaces that a grid service must implement to fit into the OGSA architecture. GT3 offers several OGSI-compliant services corresponding to usual GT2 services, such as grid resource recovery and GridFTP, and the ability to create new OGSI-compliant services. These services are provided through standard OGSI mechanisms, offering a consistent way to query any grid service about its configuration and status information.

The Grid Service Specification details how a client interacts with a grid service. OGSI software provides mandatory grid service features, such as service invocation, lifetime management, a service data interface, and security interfaces that ensure basic interoperability among all grid services. A grid service executes inside a hosting environment that supports the language the service is written in (for example, C, Java, Python, or .NET), but the hosting environment insulates grid service clients from its particular implementation language. Access to services is through the standard Web Services Definition Language. So, you can write grid service clients in any language that has bindings to WSDL. The GT3 implementation includes support for WSDL and SOAP W3C.

OGSA's development represents a natural evolution of Web services. By integrating support for transient, stateful service instances with existing Web services technologies, OGSA significantly extends the power of the Web services framework, while requiring only minor extensions to existing technologies. In the near future, an OGSA approach could fully integrate grid and Web technologies. Combining these two distributed computing paradigms could improve both well-established applications and those made possible by the support of the open grid service model.

Data and knowledge grids

Grid application areas are shifting from scientific computing toward industry and business applications. To meet those needs, data grids, an enhancement of computational grids, have been designed to store, move, and manage large data sets exploited in distributed data-intensive applications.

To advance the data grid concept, we must develop knowledge-based grids that can offer tools and environments to support data analysis, inference, and discovery in scientific and business areas.[6,7] These environments will support scientists and engineers in the implementation and use of grid-based problem-

trolled fashion, always providing a global view of the grid status and application workflow. To support structured applications composed of multiple sequential stages, VEGA offers *workspace* and *virtual resource* concepts. The workspace is an area in which users can compose objects representing resources to form a particular stage of a knowledge discovery application. The virtual resource concept lets users specify resources through constraints (such as required main memory, disk space, CPU speed, and operating system). This lets users design applications independent of the particular grid on which they'll be executed, making them reusable on different grid resources over time.

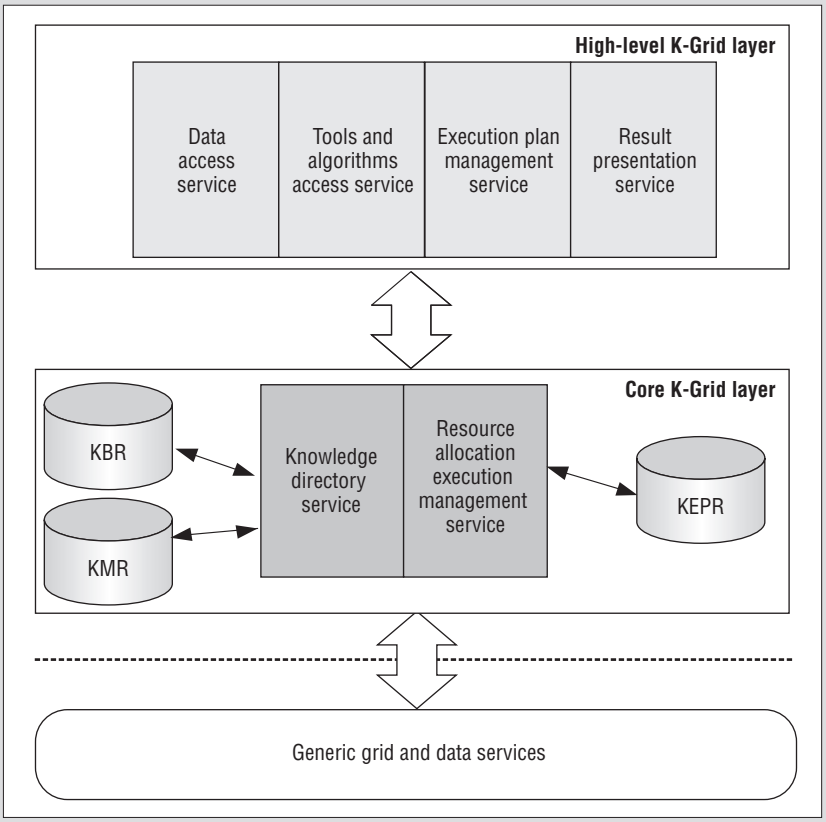

Figure A. The KNOWLEDGE GRID's main components, including repositories that provide information about resource metadata (KMR), execution plans (KEPR), and knowledge obtained as result of knowledge discovery applications (KBR).

References

1. M. Cannataro and D. Talia, "KNOWLEDGE GRID: An Architecture for Distributed Knowledge Discovery," *Comm. ACM*, vol. 46, no. 1, Jan. 2003, pp. 89–93.
2. M. Cannataro et al., "A Data Mining Toolset for Distributed High-Performance Platforms," *Proc. 3rd Int'l Conf. Data Mining 2002*, Wessex Inst. Press, 2002, pp. 41–50.

solving environments for modeling, simulation, and analysis of scientific experiments. Knowledge grids could also benefit government, industry, and commerce, where analysts must mine the large volumes of information generated by business and industrial processes to support corporate decision-making.

Recent works claimed that the creation of knowledge grids on top of computational grids is the enabling condition for developing high-performance knowledge discovery processes and meeting the challenges posed by the increasing demand for power and abstractness by complex problem-solving environments.[8,9] Knowledge grids offer high-level tools and techniques for distributed knowledge extraction from data repositories on the Grid. So, they realize the highest layer of the grid architecture (computation and data, information, and knowledge layers).

In developing the knowledge layer, the main issues are

- Synthesizing useful and usable knowledge from data
- Leveraging the Grid infrastructure to perform sophisticated data-intensive large-scale computation

To integrate knowledge discovery techniques in grid environments, we need an unambiguous representation of the knowledge base (through metadata and ontologies) that can translate moderately abstract domain-specific queries into computations and data analysis operations that can answer such queries by operating on the underlying systems. Our KNOWLEDGE GRID system offers such grid-based knowledge discovery services (see the related sidebar).

Peer to peer

P2P is at once a set of protocols, a computing model, and a design philosophy for distributed, decentralized, and self-organizing systems. In other words, P2P is a set of methodologies and technologies that lets a group of computers collaborate in a network of equals (peers), without central coordination.[10] In spite of current practices and thoughts, the Grid and P2P models share several features and have more in common than we perhaps generally recognize. Integrating the two computing models could bring many benefits in designing future scalable grids. Grids used for complex applications include a large number of nodes, so we should decentralize their functionalities to avoid bottlenecks. P2P could help ensure grid scalability: designers could use P2P to implement nonhierarchical decentralized grid systems.

The models and challenges P2P systems face are not new: peers stay at the edge of a network in which everyone creates as well as consumes—effectively, the Internet's original formulation.[11] P2P basic elements include

- Action at the edge of the network (such as computing, resource sharing, and communication)
- Shared resources between peers (such as CPU idle cycles, disk space, computing power, network bandwidth, and content)
- Direct communication between peers, which takes place without great assumptions about the underlying network and protocols (such as DNS)

P2P's core function is the management of numerous peers, usually in an unstable environment (where peers appear and disappear continuously) without central coordination. In contrast, today's grids are based on a persistent service infrastructure that often uses a central or a distributed, but hierarchical, coordination. Almost all P2P applications belong to the following categories:

- Distributed computing (such as SETI@home)
- Content sharing (such as the original Napster—this wasn't a pure form of P2P because it was based on a central directory service—and Gnutella
- Collaboration (such as instant messaging)

P2P's main potential is its ability to exploit idle computing resources, facilitating information exchange (information discovery and content distribution). From the Grid point of view, P2P's main interesting aspects are scalability, self-configuration, autonomic management, dynamic resource discovery, and fault tolerance. On the other hand, current P2P systems often lack the ability to deploy production-quality services, such as QoS negotiation, persistent and multipurpose service infrastructure, complex services (beyond simple file-sharing), robustness, performance, and security.

A synergy between P2P systems and grids will lead to new highly distributed systems in which each computer contributes to solving a problem or implementing a system while also using services offered by other computers in the network.[12] OGSA can provide opportunities to integrate P2P and the Grid. The architecture defines standard mechanisms for creating, naming, and discovering persistent and transient grid-service instances. For the next-generation grid, it'll be very interesting to determine how to deploy OGSA-oriented grid protocols to build P2P applications. By implementing service instances in a P2P manner within such a framework, developers can provide P2P service configuration and deployment on the grid infrastructure.

Pervasive and ubiquitous computing

Ubiquitous computing describes distributed computing devices, such as personal devices, wearable computers, and sensors in the environment, and the software and hardware infrastructures needed to support applications on these computing devices. The terms "pervasive" and "ubiquitous" are used interchangeably. As Mark Weiser described in his well-known article,[13] ubiquitous computing means interconnected hardware and software so pervasive and so integrated in the environment that no one notices their presence. We can expand this characterization if we consider ubiquitous computing's main dimensions:[14]

- Mobility of users, devices (PDA, phones), and software (mobile agents)
- Degree to which devices are embedded into the environment

Mobile computing is about the ability to physically move computing services with users, but the computing model doesn't change considerably while the users move. In pervasive computing, the device can obtain information from the environment in which it is embedded and adapt to its behavior—for example, by dynamically building (choosing) a suitable model of computing. When we have a combination of high mobility and high embeddedness, then any device, while moving with the user, can build incremental models of the visited environments and configure its services accordingly. Conversely, the software (the environment) can adapt itself to the currently available devices. Integrating large-scale mobility and pervasive computing functionality in grid systems poses new challenges and requirements to the underlying architecture, such as the following:

- Ontology-based semantic modeling (user preferences, devices characteristics, and context) would enable reasoning about the user's needs and required adaptation of services.
- An adaptable and composable software infrastructure could find, adapt, and deliver appropriate applications (services) to the user's computing environment (devices) on the basis of context. To execute a user task, the computing platform should dynamically find and compose the appropriate components and services; once instantiated, the application might need to move between devices and environments. An interesting approach along this direction is described elsewhere.[15]

A next-generation grid architecture

As we envision it, next-generation grids will require

- Knowledge discovery and knowledge management functionalities, for users' needs (such as intelligent exploration of data) and system management
- Semantic modeling of users' tasks and needs, grid services, data sources, computing devices (from ambient sensors to high-performance computers)
- Pervasive and ubiquitous computing, through context awareness and adaptation
- Advanced forms of collaboration, through dynamic formation of virtual organizations
- Self-configuration, autonomic management, dynamic resource discovery, and fault tolerance

In particular, among these functionalities, we believe that next-generation grids should first provide the three following main classes of services and related architectural frameworks.

Knowledge management and ontology-based services

These services are used to build, manipulate, and interoperate the grid knowledge base homogeneously. By *grid knowledge base*, we mean all data that the Grid stores, maintains, and updates for users, applications, and operations. It comprises, for example, the Globus Monitoring and Discovery Service data and metadata, grid services usage data, and application data sources and results. Grid middleware or grid applications currently maintain much of this data, so the key challenge for next-generation grids will be their seamless integration and use.

It's too soon to say whether the winning approach will be data integration at the database level or a virtual integration through the building of distributed services. We think that both approaches will be used: for example, the Globus project has recently used the former approach to implement some OGSI-compliant database services in the GT3. From an architectural point of view, technologies useful for building, manipulating, and reasoning on the grid knowledge base are ontologies, logic pro-

gramming, workflow management, and constraint programming. In such scenarios, one or more ontologies classify each "object" on the Grid (as in the Semantic Web) into the knowledge base. Two important services that could be offered are ontology-based grid programming and request-resource matchmaking.

Knowledge discovery services

These services extract knowledge from the data stored inside the grid knowledge base. We'll be able to use these services to build high-level knowledge discovery applications, such as the KNOWLEDGE GRID, and to enhance existing basic grid services. Two possible applications that require distributed data mining functionalities and accessing distributed partitions of a knowledge base are grid-based document management (that is, document classification and retrieval over the grid) and an enhanced version of the GridFTP protocol using data mining techniques to predict optimal transfer parameters.

Dynamic resource discovery and adaptation

When a grid goes beyond a static, well-established configuration and new devices and resources can dynamically enter and exit the grid, it becomes a pervasive grid. When this happens, new services that can adapt themselves to the environment have to be developed. For example, P2P technologies could implement dynamic discovery algorithms, and developers could use adaptation techniques from the Adaptive Hypermedia research community[16] to adapt services to a user's computing environment (devices) on the basis of context.

Some emerging technologies partly fulfill these requirements, and we envision that next-generation grids will be based on their integration and composition. Figure 1 shows those technologies with respect to two principal dimensions: knowledge and distribution.

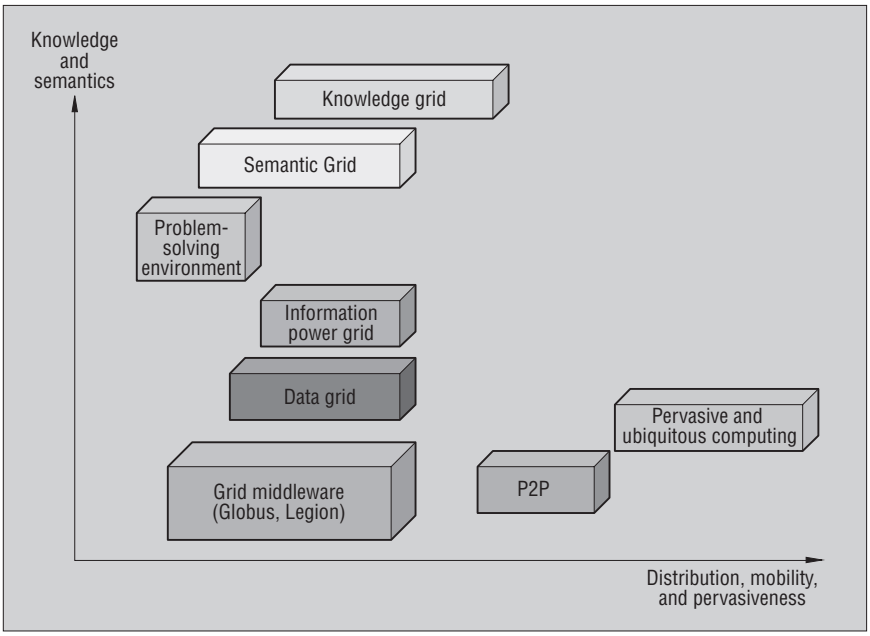

Figure 1. Key technologies for grids.

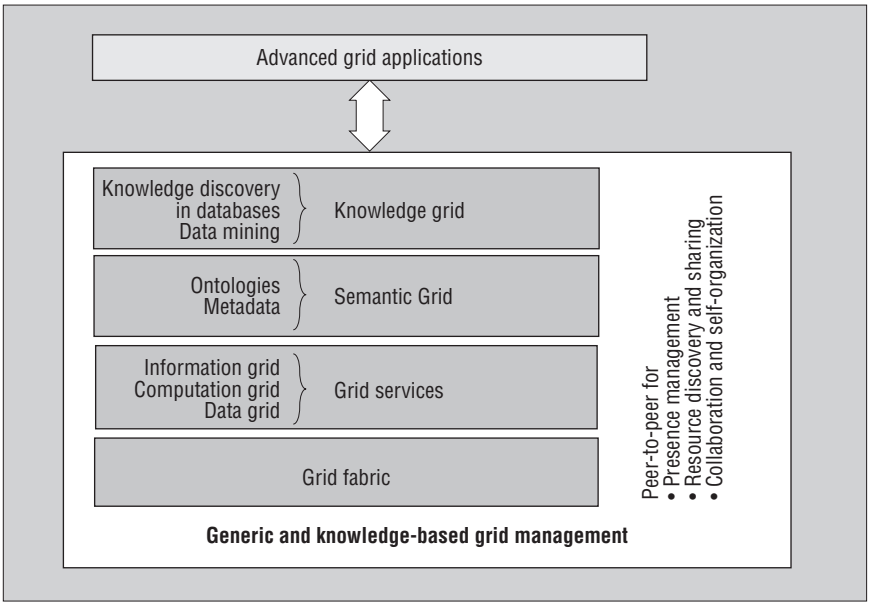

Figure 2. The next-generation grid's main layers.

Grid middleware is starting to embed more and more semantic- and knowledge-based services. Conversely, few have tried to employ P2P techniques in grids or to support pervasiveness and mobility, such as in ubiquitous systems. Unfortunately, building pervasive grids is not as easy as supporting new devices entering the systems or discovering their state in a timely manner. The increasing number of application domains supported by grids will require seamless protocols, allowing easy and quick participation in grid computations and access to grid services. Most probably that will impact different aspects of grid middleware, such as security and single sign-on, which should be more decentralized.

Figure 2 summarizes a layered general architecture for the next-generation grid we've described. The figure shows how recent research initiatives in the grid community (OGSA, the Semantic Grid, and knowledge grids) could be composed to provide a coherent architecture of services. Although these initiatives have some overlap, they complement each other.

Some of the technologies discussed, such as ontologies and reasoning, knowledge management, and knowledge discovery, are embedded in the grid initiatives cited so far, but their impact will be really evident when they're used internally to enhance grid management and operation.

P2P and ubiquitous computing techniques have started to be used very recently. In our opinion, P2P will be the orthogonal technology on which primary tasks such as presence

management, resource discovery and sharing, collaboration, and self-configuration will be based.

Pervasive grids, employing ubiquitous computing techniques, will transform the fabric of the Grid into a stable core of resources that dynamically increases and decreases whenever new resources join or leave the Grid.

A case study: DAMON

In our case study, we used semantics and knowledge to enhance grid functionalities and create new grid services. We used ontology-based semantic modeling to enhance component-based programming on the Grid.

In component-based grid programming, the user designs an application using available software components. However, the components' selection and connection are often left up to the designer. In this case study, we'll show how ontologies can help users design and program knowledge discovery applications in the KNOWLEDGE GRID.

DAMON

DAMON (*da*ta *m*ining *on*tology) is an ontology that explicitly manages knowledge about data mining and related software tools. It offers users a reference for the different classes of data mining tasks, methodologies, and software components available to solve a given problem.[17] The choice of how to structure an ontology determines what a system can know and reason about. Our ontology classifies data mining software that lets a user select the most appropriate software to solve a KDD (knowledge discovery in databases) problem. The ontology represents the data mining software's features by classifying their main components and showing their relationships and constraints. We categorized the data mining software using the following classification parameters:

- A (data mining) *task* represents a data mining technique for extracting patterns from data—that is, a task specifies a data mining process's goal.
- A *method* is a data mining methodology used to discover the knowledge; different methods serve different purposes. You can think of it as a structured manipulation of the input data to extract knowledge.
- An *algorithm* is the way in which a data mining task is performed.
- S*oftware* is an implementation (in a programming language) of a data mining algorithm.
- A *suite* implements a set of data mining algorithms: every algorithm may perform different tasks and employ different methods to achieve the goal.
- A *data source* is the input on which data mining algorithms work to extract new knowledge.
- The *human interaction* specifies how much human interaction with the discovery process is required or supported.

We can get information about data mining tasks and methodologies, and specific software components implementing data mining algorithms, by browsing or searching the ontology. In particular, we've implemented semantic search (concept-based) of data mining software and other data mining resources.

Ontology browsing. When browsing an ontology, users can navigate via different points of access, showing deeper levels of details. For example, a user starting at the top of the ontology could navigate toward more specific topics by clicking the classes of interest (diving into the information).

Ontology-based semantic search. For this type of search, a user might query very detailed information about data mining resources annotated in DAML+OIL. The result set is very accurate because concepts from the underlying ontology clearly indicate the searched terms' semantic content. Our ontology-based search engine supports several kinds of simple inference that can broaden queries, including equivalence, inversion, generalization, and specialization. For example, if the query result set is empty, the user can at least find objects that partially satisfy the query: some classes can be replaced by their superclasses or subclasses. We can both narrow and broaden the query's scope because of the domain description's ontological nature.

Moreover, in addition to finding out where and how to access the available data mining software, DAMON can help a user by searching all the available software that satisfies some user requirements, such as performing a given task (such as classification), implementing a given algorithm (such as CHAID, *chi*-squared *a*utomatic *i*nteraction *d*etector), using a specific methodology (such as decision trees), and requiring a specific input format. Some possible DAMON queries are

- Find data sources about a specific topic.
- Find software implementing a desired data mining algorithm.
- Find software performing a specified data mining task.
- Find software or an algorithm using particular methods.

Our ontology can also help the user with query formulation. Users encounter difficulties when they need to provide terms that best describe their information need (vocabulary problem). DAMON explicitly shows the classes that describe the domain of interest, simplifying the vocabulary choice.

Ontology-based grid programming

A KNOWLEDGE GRID application designer can use DAMON as an ontology-based assistant—which suggests what to do and use based on the user's needs—and as a tool for semantic search of data mining software. In other words, it could help both novice and expert KNOWLEDGE GRID users enhance an application's formulation and design, assisting with the selection and configuration of the most suitable data mining solution for a specific KDD process or providing specific details on the different data mining resources.

The data mining knowledge base we use to support knowledge discovery programming has two conceptual layers: at the top layer, DAMON gives general information about the data mining domain; specific information about installed software components and data sources are maintained where the resources reside. From an architectural point of view, the ontology is a central resource, whereas specific metadata are distributed resources. As an example, DAMON stores the information that the C5.0 software implements the C5 algorithm, which uses the decision tree method, a classification method. The ontology's `C5.0` software node contains the URLs of the metadata files describing details about all of the software's installed instances (including technical parameters, availability, location and configuration of data mining software and tools).

For example, a user logged on the KNOWLEDGE GRID node `KU` needs to run a data mining application composed of two tasks, clustering and classification, on the data set `DBX` stored on the KNOWLEDGE GRID node `KD`. The user needs to cluster the data set using three different algorithms running in parallel on a copy of the data set. Clustering results must be analyzed by a classification algorithm that will be executed in parallel on three different nodes, generating three classification

models of the same data set.

Using DAMON, the user first searches the clustering algorithms by browsing or querying the ontology based on some user requirements (such as the algorithm's computational complexity, attitude to solve the given problem, or the method used to perform the data mining). Then, she searches the clustering software implementing the algorithms and working on the data set DBX and locates the metadata URLs referring to the nodes KG1, KG2, and KG3, offering the clustering software K-Means, Intelligent Miner, and AutoClass. Moreover, the user also finds the node KG4 that offers the C5.0 classifier. At this point, the user can access specific information about the software by accessing the specific metadata on each identified node.

The obtained information is then used for the visual composition of those software components and data sources through VEGA's graphic interface. The KNOWLEDGE GRID translates this component-based application's abstract description into the grid submission language (the Resource Specification Language, in the Globus case). After application execution and result collection, the user can enter the implemented application into the DAMON ontology. In this way, users can enrich and extend the knowledge base with new, complex data mining tasks.

The programming example we've discussed is just one way you can use knowledge and semantic grid services for high-level programming of complex applications on grids. This approach allows the reuse of previously developed applications and software components that you can integrate in new grid applications.

To achieve a pervasive, worldwide knowledge management infrastructure, next-generation grids should include knowledge discovery and knowledge management functionalities, for both applications and system management. An emerging research field known as *grid intelligence* is exploring the way in which we can effectively acquire, represent, exchange, integrate, and convert data and information available at different levels of the Grid into useful knowledge. To achieve this, we'll need to explore integrating data management and artificial intelligence techniques with grid computing. ❐

The Authors

Mario Cannataro is an associate professor of computer engineering at University Magna Græcia of Catanzaro and a cofounder of Exeura. His research interests include grid computing, bioinformatics and proteomics, grid-based problem-solving environments, and adaptive hypermedia systems. He received his Laurea cum laude in computing engineering from the University of Calabria, Italy. He is a member of the IEEE Computer Society and ACM. Contact him at University Magna Græcia of Catanzaro, Via T. Campanella 115, 88100 Catanzaro, Italy; cannataro@unicz.it.

Domenico Talia is a professor of computer science at the University of Calabria, a research associate at ICAR-CNR, and a partner of Exeura. His research interests include grid computing, knowledge discovery, parallel computing and distributed systems, complex systems, and peer-to-peer computing. He is a member of the IEEE Computer Society and ACM. Contact him at the DEIS, University of Calabria, Via P. Bucci 41c, 87036 RENDE, CS, Italy; talia@deis.unical.it; wwwinfo.deis.unical.it/~talia.

Acknowledgments

Project FIRB GRID.IT, funded by MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca), partially supported this work. We thank those working in the knowledge grid research team: Carmela Comito, Antonio Congiusta, Carlo Mastroianni, Andrea Pugliese, Paolo Trunfio, and Pierangelo Veltri.

References

1. Evolving Data Mining into Solutions for Insights (special issue), *Comm. ACM*, vol. 45, no. 8, August 2002.
2. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 279, no. 5, May 2001, pp. 34–43.
3. D. De Roure, N.R. Jennings, and N. Shadbolt, "The Semantic Grid: A Future e-Science Infrastructure," *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, A.J.G. Hey, and G. Fox, eds., John Wiley & Sons, 2003, pp. 437–470.
4. I. Foster et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Globus Project, 22 June 2002; www.globus.org/research/papers/ogsa.pdf.
5. D. Talia, "The Open Grid Services Architecture: Where the Grid Meets the Web," *IEEE Internet Computing*, vol. 6, no. 6, Nov./Dec. 2002, pp. 67–71.
6. R.W. Moore, "Knowledge-Based Grids: Two Use Cases," GGF-3 Meeting, Sept. 2001; http://www-itg.lbl.gov/GPA/Moore.GGF-3.pdf.
7. R. Moore, *Knowledge-Based Grids*, tech. report SDCS TR-2001-2, San Diego Supercomputer Center, San Diego, Calif., 2001.
8. F. Berman, "From TeraGrid to Knowledge Grid," *Comm. ACM*, vol. 44, no. 11, Nov. 2001, pp. 27–28.
9. W.E. Johnston, "Computational and Data Grids in Large-Scale Science and Engineering," *Future Generation Computer Systems*, vol. 18, no. 8, Oct. 2002, pp. 1085–1100.
10. D. Barkai, "Technologies for Sharing and Collaborating on the Net," *Proc. 1st Int'l Conf. Peer-to-Peer Computing* (P2P 01), IEEE CS Press, 2001, pp. 13–28.
11. D. Schoder and K. Fischbach, "Peer-to-Peer Prospects," *Comm. ACM*, vol. 46, no. 2, Feb. 2003, pp. 27–29.
12. D. Talia and P. Trunfio, "Toward a Synergy Between P2P and Grids," *IEEE Internet Computing*, vol. 7, no. 4, July/Aug. 2003, pp. 94–96.
13. M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, Sept. 1991, pp. 66–75.
14. K. Lyytinen and Y. Yoo, "Issues and Challenges in Ubiquitous Computing," *Comm. ACM*, vol. 45, no. 12, Dec. 2002, pp. 62–65.
15. J. Blythe et al., "Transparent Grid Computing: A Knowledge-Based Approach," *Proc. 15th Innovative Applications of Artificial Intelligence Conference* (IAAI 03), AAAI Press, 2003.
16. The Adaptive Web (special issue), *Comm. ACM*, vol. 45, no. 5, May 2002.
17. M. Cannataro and C. Comito, "A Data Mining Ontology for Grid Programming," *Proc. 1st Int'l Workshop Semantics in Peer-to-Peer and Grid Computing* (SemPGrid 03), 2003, pp. 113–134; www.isi.edu/~stefan/SemPGRID.