# Artificial Intelligence and Grids: Workflow Planning and Beyond

**Yolanda Gil, Ewa Deelman, Jim Blythe, Carl Kesselman, and Hongsuda Tangmunarunkit,** *USC Information Sciences Institute*

*A key challenge for grid computing is creating large-scale, end-to-end scientific applications that draw from pools of specialized scientific components to derive elaborate new results. Given a user's high-level specification of desired results, the Pegasus system can generate executable grid workflows using AI planning techniques.*

Grid computing is emerging as a key enabling infrastructure for a wide range of disciplines in science and engineering, including astronomy, high-energy physics, geophysics, earthquake engineering, biology, and global climate change (see the "Grid Computing" sidebar).[1–3] By providing fundamental mechanisms for resource discovery, management, and sharing, grids let geographically distributed teams form dynamic, multi-institutional *virtual organizations* whose members use shared community and private resources to collaborate on solutions to common problems. This gives scientists tremendous connectivity across traditional organizations and fosters cross-disciplinary, large-scale research. Grids' most tangible impact to date could be the seamless integration of and access to high-performance computing resources, large-scale data sets, and instruments that form the basis of advanced scientific discovery. However, scientists now pose new challenges that will require the current grid computing paradigm to shift significantly.

First, science could progress significantly via the synthesis of models, theories, and data contributed across disciplines and organizations. The challenge is to enable on-demand synthesis of large-scale, end-to-end scientific applications that draw from pools of specialized scientific components to derive elaborate new results. Consider, for example, a physics-related application for the Laser Interferometer Gravitational-Wave Observatory (LIGO),[4] where instruments collect data that scientists must analyze to detect the gravitational waves predicted by Einstein's theory of relativity (see the "Searching for Gravitational Waves" sidebar). To do this, scientists run pulsar searches in certain areas of the sky for a certain time period. The observations are processed through Fourier transforms and frequency range extraction software. The analysis could involve composing a workflow comprising hundreds of jobs and executing them on appropriate grid computing resources. This might span several days and necessitate failure handling and reconfiguration to handle the dynamics of the grid execution environment.

Second, we can significantly multiply the impact of scientific research if we broaden the range of applications that it can support beyond science-related uses. The challenge is to make these complex scientific applications accessible to the many potential users outside the scientific community. In earthquake science, for example, integrated earth sciences research for complex probabilistic seismic hazard analysis can have greater impact, especially when it can help mitigate the effects of earthquakes in populated areas. In this case, users might also include safety officials, insurance agents, and civil engineers, who must evaluate the risk of earthquakes of certain magnitude at potential sites. A clear need exists to isolate end users from the complex requirements necessary for setting up earthquake simulations and executing them seamlessly over the Grid.

We are developing Pegasus, a planning system we've integrated into the Grid environment that takes a user's highly specified desired results, generates valid workflows that take into account available resources, and submits the workflows for execution on the Grid. We are also beginning to extend it as a more distributed and knowledge-rich architecture.

## Challenges for robust workflow generation and management

To develop scalable, robust mechanisms that address the complex grid applications that the scien-


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

## Grid Computing

Grid computing promises to be the solution to many of today's science problems by providing a rich, distributed platform for large-scale computations, data, and remote resource management. The Grid lets scientists share disparate and heterogeneous computational, storage, and network resources as well as instruments to achieve common goals. Although Grid resources often span across organizational boundaries, Grid middleware is built to let users easily and securely access them.

The current de facto standard in Grid middleware is the Globus Toolkit. The toolkit provides fundamental services to securely locate, access, and manage distributed shared resources. Globus Information services facilitate the discovery of available resources. Resource management services provide mechanisms for users and applications to schedule jobs onto the remote resources, as well as a means to manage them. Security is implemented using the Grid Security Infrastructure, which is based on the public key certificates. Scientists can use the Globus data management services, such as the Replica Location Service and GridFTP, to securely and efficiently locate and transfer data in a wide area.

Many projects worldwide are deploying large-scale Grid infrastructure. Projects in the US include the International Virtual Data Grid Laboratory (iVDGL), the Particle Physics Data Grid (PPDG), and the Teragrid. In Europe projects such as the LHC Computing Grid Project (LCG), the Enabling Grids for E-Science in Europe (EGEE) initiative, and projects under the UK e-Science program are building the necessary infrastructure for providing a platform for scientists from disciplines such as physics, astronomy, earth sciences, biology, and so on.

Although the basic Grid building blocks are widely used, higher-level services dealing with application-level performance and distributed data and computation management are still under research and development. Among US projects addressing such issues are the Grid Physics Network (GriPhyN) project, the National Virtual Observatory (NVO), Earth System Grid (ESG), the Southern California Earthquake Center (SCEC) ITR project, and others. In Europe, much research is being carried within the UK e-science projects, the EU GridLab project, and others.

Grid computing is undergoing a fundamental change—it's shifting toward the Web Services paradigm. Web Services define a technique for describing accessible software components (that is, services), methods for discovering them, and protocol for accessing them. Grid services extend Web Services models and interfaces to support distributed state management. Among the necessary extensions are the ability to manage transient services and their lifetimes and the ability to introspect the services' characteristics and states. Grid services can be dynamically created and destroyed. Web services, and therefore Grid services, are neutral to programming language, programming model, and system software.

Another important aspect of Grid services is the support they are receiving from the wide Grid community. Meetings such as the Global Grid Forum bring together a broad spectrum of researchers and developers from academia and industry with the goal of sharing ideas and standardizing interfaces.

The tremendous advances in Grid computing research are possible due to international collaboration and the financial support of several funding agencies, the National Science Foundation, the Department of Energy, the National Aerospace Agency, and others in the US as well as the European Union and the UK government, and governments in Asia and Australia.

For more information about the Grid and related projects, please refer to the following publications and Web sites:

- I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999
- I. Foster et al., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. High Performance Computing Applications*, vol. 15, 2001, pp. 200–222
- I. Foster et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Globus Project, 2002; www.globus.org/research/papers/ogsa.pdf
- I. Foster et al., "Grid Services for Distributed System Integration," *Computer*, vol. 35, no. 6, June 2002, pp. 37–46
- Global Grid Forum: www.globalgridforum.org
- The Globus Project: www.globus.org
- Enabling Grids for E-science and Industry in Europe: http://egee-intranet.web.cern.ch/egee-intranet/gateway.html
- Earth Systems Grid: www.earthsystemgrid.org
- The Grid Physics Network project: www.griphyn.org
- International Virtual Data Grid Laboratory: www.ivdgl.org
- Large Hadron Collider Computing Grid Project: http://lcg.web.cern.ch/LCG
- National Virtual Observatory: www.us-vo.org
- Particle Physics Data Grid: www.ppdg.net
- Southern California Earthquake Center: www.scec.org

tific community envisions, we need expressive and extensible ways of describing the Grid at all levels. We also need flexible mechanisms to explore trade-offs in the Grid's complex decision space that incorporate heuristics and constraints into that process. In contrast, Grids today use syntax or schema-based resource matchmakers, algorithmic schedulers, and execution monitors for scripted job sequences, which attempt to make decisions with limited information about a large, dynamic, and complex decision space. Clearly, a more flexible and knowledge-rich Grid infrastructure is needed. Specifically, we must address the following issues.

### Knowledge capture

High-level services such as workflow generation and management systems are starved for information and lack expressive descriptions of grid entities and their relationships, capabilities, and trade-offs. Current grid middleware simply doesn't provide the expressivity and flexibility necessary for making sophisticated planning and scheduling decisions. Something as central to the Grid as resource descriptions are still based on rigid schemas. Although higher-level middleware is under development,[2,5] grids will have a performance ceiling determined by limited expressivity and the amount of information and knowledge available for making intelligent decisions.

### Usability

Exploiting distributed heterogeneous

## Searching for Gravitational Waves

One application of the Pegasus workflow planning system (see http://Pegasus.isi.edu)[1–5] helps analyze data from the Laser Interferometer Gravitational-Wave Observatory project. LIGO is the largest single enterprise the US National Science Foundation has undertaken to date. It aims to detect gravitational waves that, although predicted by Einstein's theory of relativity, have never been observed experimentally. By simulating Einstein's equations, scientists predict that those waves should be produced by colliding black holes, collapsing supernovae, pulsars, and possibly other celestial objects. With facilities in Livingston, Louisiana, and Hanford, Washington, LIGO is joined by gravitational-wave observatories in Italy, Germany, and Japan to search for these signals.

The Pegasus planner that we've developed is a tool that scientists can use to analyze the data LIGO collects. In fall 2002, scientists held an initial 17-day data collection effort, followed by a two-month run in February 2003, with additional runs to be held throughout the project's duration. We used Pegasus with LIGO data collected during the instrument's first scientific run, which targeted a set of locations of known pulsars as well as random locations in the sky. Pegasus generated end-to-end grid job workflows that ran on computing and storage resources at Caltech, the University of Southern California, the University of Wisconsin, the University of Florida, and the National Center for Supercomputing Applications. It scheduled 185 pulsar searches with 975 tasks, for a total runtime of almost 100 hours on a grid with machines and clusters with different architectures.

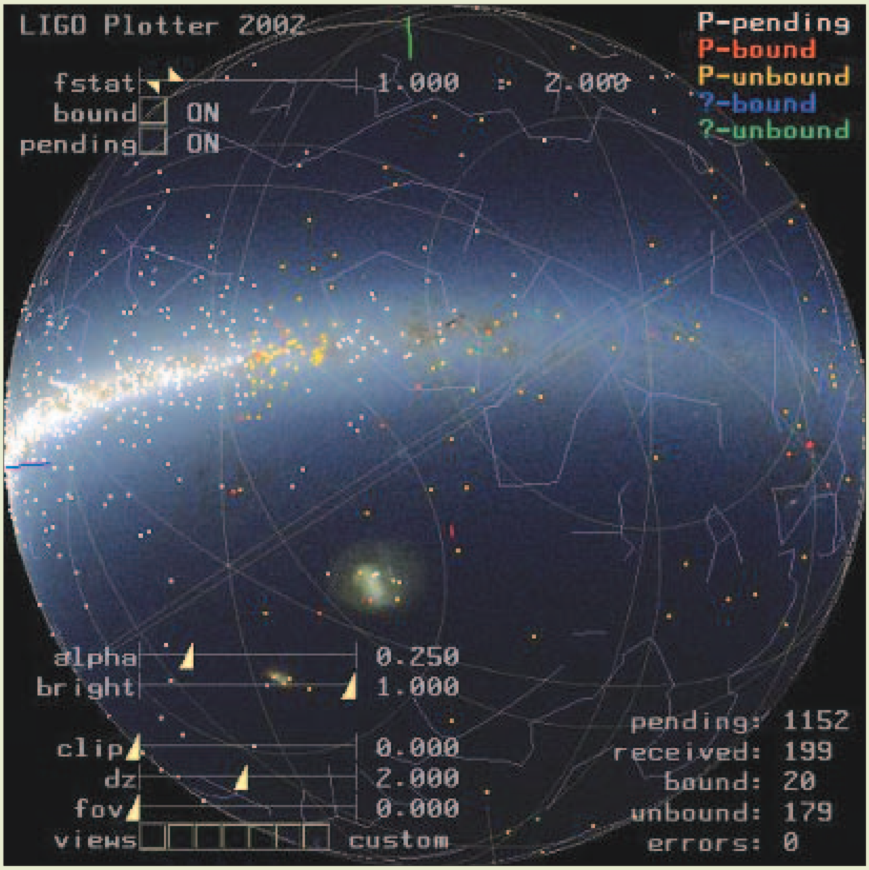



**Figure A. Visualization of a Laser Interferometer Gravitational-Wave Observatory pulsar search. The sphere depicts the map of the sky. The points indicate the locations where the search was conducted. The color of the points indicates the range to which particular results belong.**

Figure A illustrates the results of a pulsar search done with Pegasus. Scientists specify the search ranges through a Web interface. The figure's top left corner shows the specific range displayed in this visualization. The bright points represent the locations searched. The red points are pulsars within the bounds specified for the search, and the yellow ones are pulsars outside those bounds. The blue and green points are the random points searched, within and outside the bounds, respectively.

Pegasus demonstrates the value of planning and reasoning with declarative representations of knowledge about various aspects of grid computing, such as resources, application components, users, and policies, which are available to several different modules in a comprehensive workflow tool for grid applications. As the LIGO instruments are recalibrated and set up to collect additional data in the coming years, Pegasus will confront increasingly challenging workflow generation tasks as well as grid execution environments.

### References

1. E. Deelman et al., "Workflow Management in GriPhyN," *Grid Resource Management*, J. Nabrzyski, J. Schopf, and J. Weglarz, eds., Kluwer, 2003.
2. E. Deelman et al., "Pegasus: Mapping Scientific Workflows onto the Grid," to be published in *Proc. 2nd European Across Grids Conf.*, 2004.
3. E. Deelman et al., "Mapping Abstract Complex Workflows onto Grid Environments," *J. Grid Computing*, vol. 1, no. 1, 2003, pp. 25–39.
4. J. Blythe et al., "The Role of Planning in Grid Computing," *Proc. Int'l Conf. Automated Planning and Scheduling* (ICAPS), 2003.
5. J. Blythe et al., "Transparent Grid Computing: A Knowledge-Based Approach," *Proc. 15th Innovative Applications of Artificial Intelli-*

resources is already difficult, much more so when it involves different organizations with specific usage policies and contentions. All these mechanisms must be managed, and, sadly, today that burden falls on the end users. Even though users think in much more abstract, application-level terms, today's grid users must have extensive knowledge of the Grid computing environment and its middleware functions.

For example, a user must know how to find the physical locations of input data files through a replica locator. He or she must understand the different types of job schedulers running on each host and their suitability for certain types of tasks, and the user must consult access policies to make valid resource assignments that often require resolving denial of access to critical resources. Users should be able to submit high-level requests in terms of their application domain.

Grids should provide automated workflow generation techniques that would incorporate the necessary knowledge and expertise to access grids while making more appropriate and efficient choices than the users themselves would. The challenge of usability is key because it is insurmountable for many potential users who currently shy away from grid computing.

### Robustness

Failures in highly distributed heterogeneous systems are common. The Grid is a dynamic environment where the resources are highly heterogeneous and shared among many users. Failures can be common hardware and software failures but can also result from other modes. For example, a resource usage policy might change, making the resource effectively unavailable.

Worse, while the execution of many workflows spans days, they incorporate submission information that's doomed to change in a dynamic environment such as the Grid. Users must provide details about which replica of the data to use or where to submit a particular task, sometimes days in advance. The user's choices at the execution's beginning might not yield good performance further in the run.

Even worse, the underlying execution system might have changed so significantly (owing to failure or resource usage policy change) that the execution can no longer proceed. Without knowing the workflow execution history—the underlying reasons for making particular refinement and scheduling decisions—users may be unable to rescue the execution.

Grids need more information to ensure proper completion, including knowledge about workflow history, the current status of their subtasks, and the decisions that led to their particular design. The gains in efficiency and robustness of execution in this more flexible environment, especially as applications scale in size and complexity, could be enormous.

### Access

The Grid's multiorganizational nature makes access control important and complex. The resources must be able to handle users from different groups, who probably have different access and usage privileges. Grids provide an extremely rich, flexible basis for approaching this problem through authentication, security, and access policies both at the user and organization levels. Today's resource brokers schedule tasks on the Grid and give preference to jobs on the basis of their predefined policies and those of the resources they oversee. But as the Grid supports larger and more numerous organizations and users become more differentiated (consider the needs of students versus those of scientists, for example), these brokers will need to consider complex policies and resolve conflicting requests from the Grid's many users. New facilities will be necessary for supporting advance reservations to guarantee availability and provisioning of additional resources for anticipated needs. Without a knowledge-rich infrastructure, fair and appropriate use of grid environments will be impossible.

While the execution of many workflows spans days, they incorporate submission information that's doomed to change in a dynamic environment such as the Grid.

### Scale

Today, typical scientific grid applications run over days or weeks and process terabytes of data. In the near future, the data will reach the petabyte scale. Even the most optimized application workflows risk performing poorly when they execute. Such workflows are also fairly likely to fail owing to simple circumstances, such as a lack of disk space. Large amounts of data are only one characteristic of such applications. The scale of the workflows themselves also contributes to the problem's complexity. To perform a meaningful scientific analysis, hundreds of thousands of workflows might need execution. These might be coordinated to result in more efficient, cost-effective grid usage. A need exists for managing complex workflow pools that balance access to resources, adapt the execution of the application workflows to exploit newly available resources, provide or reserve new capabilities if the foreseeable resources aren't adequate, and repair the workflows in case of failure. Such a framework could enable enormous scientific advances.

## Pegasus: Generating executable grid workflows

Our focus to date has been workflow composition as an enabling technology that can publish components and compile them into an end-to-end workflow of jobs for execution on the Grid. We've used AI planning techniques, where the alternative component combinations are formulated in a search space with heuristics that represent the complex trade-offs that arise in grids.

Our workflow generation and mapping system, Pegasus,[6,7] integrates an AI planning system into a grid environment. In one Pegasus configuration, a user submits an application-level description of the desired data product. The system then generates a workflow by selecting appropriate application components, assigning the required computing resources, and overseeing the successful execution. The workflow can be optimized on the basis of the estimated runtime. We tested the system in two different gravitational-wave physics applications, where it generated complex workflows of hundreds of jobs and submitted them for execution on the Grid over several days.[8]

We cast the workflow generation problem as an AI planning one in which the goals are the desired data products and the operators are the application components.[9,10] An AI planning system typically receives as input

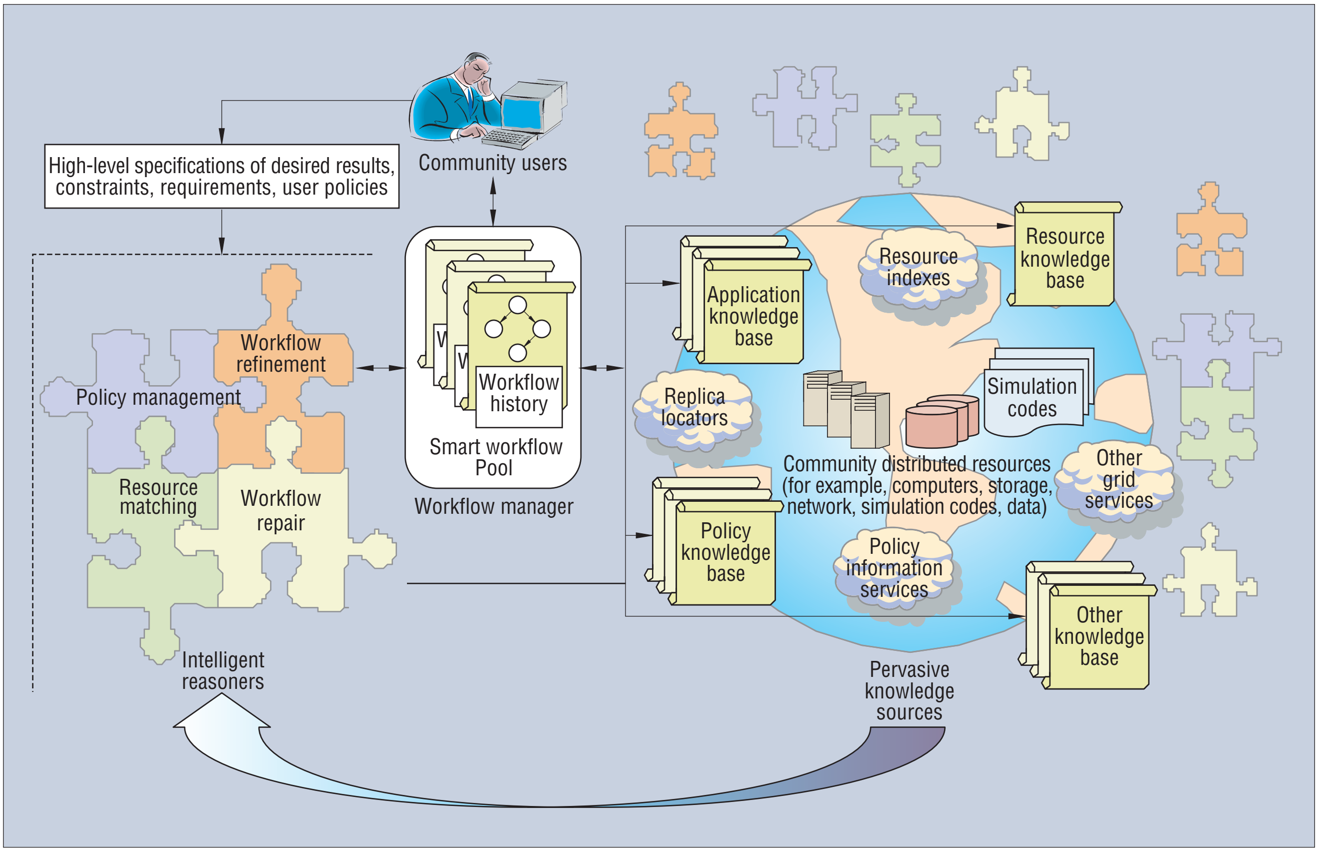



**Figure 1. Distributed grid workflow reasoning.**

a representation of the current state of its environment, a declarative representation of a goal state, and a library of operators that the planner can use to change the state. Each operator has a description of the states in which it can legally be used, called *preconditions*, and a concise description of the changes to the state that will take place, called *effects*. The planning system searches for a valid, partially ordered set of operators that will transform the current state into one that satisfies the goal. Each operator's parameters include the host where the component is to run, while the preconditions include constraints on feasible hosts and data dependencies on required input files. The plan returned corresponds to an executable workflow, which includes the assignment of components to specific resources that can be executed to provide the requested data product.

The declarative representation of actions and search control in domain-independent planners is convenient for representing constraints such as computation and storage resource access and usage policies. Planners can also incorporate heuristics, such as preferring a high-bandwidth connection between hosts performing related tasks. Additionally, planning techniques can provide high-quality solutions, partly because they can search several solutions and return the best ones found, and because they use heuristics that will likely guide the search to good solutions.

Pegasus takes a request from the user and builds a goal and relevant initial state for the AI planner, using grid services to locate relevant existing files. Once the plan is complete, Pegasus transforms it into a directed acyclic graph to pass to DAGMan[11] for execution on the Grid.

We are using Pegasus to generate executable grid workflows in several domains,[7] including genomics, neural tomography, and particle physics (see the LIGO application description in the "Searching for Gravitational Waves" sidebar).

As we attempt to address more aspects of the grid environment's workflow management problem, including failure recovery, respecting institutional and user policies and preferences, and optimizing various global measures, we find that, as mentioned, a more distributed and knowledge-rich approach is required.

## Future grid workflow management

We envision many distributed heterogeneous knowledge sources and reasoners, as Figure 1 shows. The current grid environment contains middleware that can find components that can generate desired results, find the input data they require, find replicas of component files in specific locations, match component requirements with available resources, and so on. This environment should be extended with expressive declarative representations that capture currently implicit knowledge, and should be available to various reasoners distributed throughout the Grid.

In our view, workflow managers will coordinate the generation and execution of workflow pools. The workflow managers' main responsibilities are to

- Oversee their assigned workflows' development and execution
- Coordinate among workflows that might have common subtasks or goals
- Apply fairness rules to ensure that workflows execute in a timely manner

The workflow managers also identify reasoners that can refine or repair the workflows as needed. You can imagine deploying a workflow manager per organization, per type of workflow, or per group of resources, whereas the many knowledge structures and reasoners will be independent from this mode of deployment. The issue of workflow coordination is particularly crucial in some applications where significant savings result from reusing data products from current or previously executed workflows.

Users provide high-level specifications of desired results and, possibly, constraints on the components and resources to be used. The user could, for example, request that the system conduct a pulsar search on data collected over a given time period. The user could constrain the request further by stating a preference for using Teragrid resources or certain application components with trusted provenance or performance. These requests and preferences will be represented declaratively and made available to the workflow manager. They will form the initial smart workflow. The reasoners that the workflow manager indicates will then interpret and progressively work toward satisfying the request.

In the case just mentioned, workflow generation reasoners would invoke a knowledge source with descriptions of gravitational-wave physics applications to find relevant application components. They would refine the request by producing a high-level workflow comprising these components. The refined workflow would contain annotations about the reason for using a particular application component and indicate the source of information used to make that decision. As the workflow is being refined, additional steps can be added to satisfy the user's requirements and to process the data for other existing steps.

At any given time, the workflow manager can be responsible for numerous workflows in various stages of refinement. The tasks in a workflow needn't be homogeneously refined as it develops but can have different degrees of detail. Some reasoners will specialize in tasks in a particular development stage—for example, a reasoner that performs the final assignment of tasks to the resources will consider only tasks in the smart workflow that are "ready to run."

The reasoners will generate workflows with executable portions and partially specified portions, and iteratively add details on the basis of the execution of their initial portions and the current state of the execution

## Related Work

Although scientists naturally specify application-level, science-based requirements, the Grid dictates that they make quite prosaic decisions (for example, which data replica to use, where to submit a particular task, and so on). Also, they must oversee workflow execution, often over several days, when changes in use policies or resource performance could render the original job workflows invalid.

Recent grid projects focus on developing higher-level abstractions to facilitate composing complex workflows and applications from a pool of underlying components and services, such as the GriPhyN Virtual Data Toolkit[1] and the GrADS dynamic application configuration techniques.[2] The GriPhyN project is developing catalogs, planners, and execution environments to enable the virtual data concept, as well as the Chimera system[3] for provenance tracking and virtual data derivation. These projects don't emphasize automated application-level workflow generation, execution repair, or optimization. The International Virtual-Data Grid Laboratory[4] also centers on data management uses of workflows and doesn't address automatic workflow generation and management. The GrADS project has investigated dynamic application configuration techniques that optimize application performance based on performance contracts and runtime configuration. However, these techniques are based on schema-based representations that provide limited flexibility and extensibility and algorithms with complex program flows to navigate through that schema space.

myGrid is a large, ongoing UK-funded project that provides a scientist-centered environment to data management for grid computing. It shares with our approach the use of a knowledge-rich infrastructure that exploits ontologies and Web services. Some research is investigating semantic representations of application components using semantic markup languages such as DAML-S,[5] and exploiting DAML+OIL and description logics and inference to support resource matchmaking and discovery. Our work is complementary in that myGrid doesn't include reasoners for automated workflow generation and repair.

Researchers have used AI planning techniques to compose software components[6,7] and Web services.[8,9] However this work doesn't address key areas for Grid computing such as allocating resources for higher quality workflows and maintaining the workflow in a dynamic environment.

### References


1. P. Avery and I. Foster, *The GriPhyN Project: Towards Petascale Virtual Data Grids*, tech. report GriPhyN-2001-15, Grid Physics Network, 2001.
2. F. Berman et al., "The GrADS Project: Software Support for High-Level Grid Application Development," *Int'l J. High-Performance Computing Applications*, vol. 15, 2001.
3. J. Annis et al., *Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey*, tech. report GriPhyN-2002-05, Grid Physics Network, 2002.
4. P. Avery, *An International Virtual-Data Grid Laboratory for Data Intensive Science*, tech. report GriPhyN-2001-2, Grid Physics Network, 2001.
5. A. Ankolekar et al., "The DAML Services Coalition: DAML-S: Web Service Description for the Semantic Web," *Proc. 1st Int'l Semantic Web Conf.* (ISWC), 2002.
6. S.A. Chien and H.B. Mortensen, "Automating Image Processing for Scientific Data Analysis of a Large Image Database," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, Aug. 1996, pp. 854–859.
7. K. Golden, "Universal Quantification in a Constraint-Based Planner," *Proc. 6th Int'l Conf. Artificial Intelligence Planning Systems* (AIPS), AAAI Press, 2002.
8. S. McIlraith and T. Son, "Adapting Golog for Composition of Semantic Web Services," *Proc. 8th Int'l Conf. Principles of Knowledge Representation and Reasoning* (KR 2002), 2002, pp. 482–496.
9. D. McDermott, "Estimated-Regression Planning for Interactions with Web Services," *Proc. 6th Int'l Conf. Artificial Intelligence Planning Systems* (AIPS), AAAI Press, 2002.

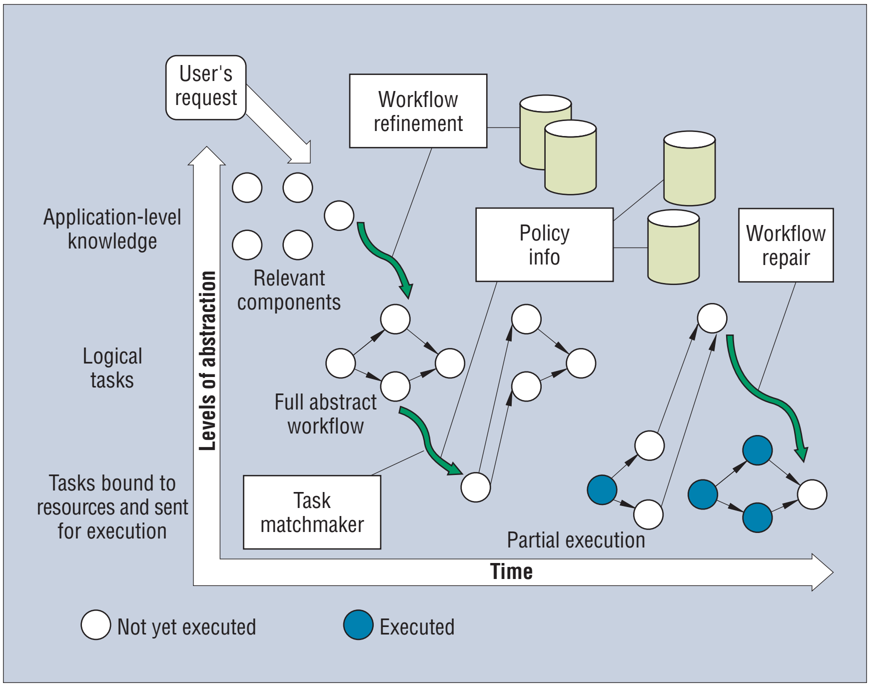



**Figure 2. Workflows are incrementally refined over time by distributed reasoners.**

environment (see Figure 2). Users can find out at any time the workflow's status and can modify or guide the refinement process if they desire. For example, users can reject particular choices the reasoner makes regarding application components and can incorporate additional preferences or priorities.

Knowledge sources and intelligent reasoners should be accessible as *grid services*,[12] the widely adopted new grid infrastructure supported by the recent implementation of the Open Grid Services Architecture. Grid services build on Web Services and extend them with mechanisms to support distributed computation. For example, Grid services offer subscription and update notification functions that facilitate the handling of the dynamic nature of Grid information. They also offer guarantees of service delivery through service-versioning requirements and expiration mechanisms. Grid services are also implemented on scalable, robust mechanisms for service discovery and failure handling. The Semantic Web, semantic markup languages, and other technologies, such as Web Services,[13,14] offer critical capabilities for our vision.

More declarative, knowledge-rich representations of computation and problem solving will result in a globally connected information and computing infrastructure that will harness the power and diversity of massive amounts of online scientific resources. Our work contributes to this vision by addressing two central questions:

- What mechanisms can map high-level requirements from users into distributed executable commands that pull numerous distributed heterogeneous services and resources with appropriate capabilities to meet those requirements?
- What mechanisms can manage and coordinate the available resources to enable efficient global use and access given the scale and complexity of the applications possible with this highly distributed heterogeneous infrastructure?

The result will be a new generation of scientific environments that can integrate diverse scientific results and whose sum will be orders of magnitude more powerful than their individual ingredients. The implications will go beyond science and into the realm of the Web at large. ■

## Acknowledgments

We thank Gaurang Mehta, Gurmeet Singh, and Karan Vahi for developing the Pegasus system. We also thank Adam Arbree, Kent Blackburn, Richard Cavanaugh, Albert Lazzarini, and Scott Koranda. The visualization of LIGO data was created by Marcus Thiebaux using a picture from the Two Micron All Sky Survey NASA collection. This research was supported partly by the National Science Foundation under grants ITR-0086044 (GriPhyN) and EAR-0122464 (SCEC/ITR), and partly by an internal grant from the Information Sciences Institute.

## References

1. J. Annis et al., *Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey*, tech. report GriPhyN-2002-05, Grid Physics Network, 2002; www.griphyn.org/chimera/papers/SDSS-SC2002.crc.submitted.pdf.

2. P. Avery and I. Foster, *The GriPhyN Project: Towards Petascale Virtual Data Grids*, tech. report GriPhyN-2001-15, Grid Physics Network, 2001.

3. E. Deelman et al., "GriPhyN and LIGO, Building a Virtual Data Grid for Gravitational Wave Scientists," *Proc. 11th Int'l Symp. High-Performance Distributed Computing*, IEEE CS Press, 2002.

4. A. Abramovici et al., "LIGO: The Laser Interferometer Gravitational-Wave Observatory (in Large Scale Measurements)," *Science*, vol. 256, 1992, pp. 325–333.

5. T.H. Jordan et al., "The SCEC Community Modeling Environment—An Information Infrastructure for System-Level Earthquake Research," *Seismological Research Letters*, vol. 74, no. 3, May/June 2003, pp. 44–46.

6. E. Deelman et al., "Workflow Management in GriPhyN," *Grid Resource Management*, J. Nabrzyski, J. Schopf, and J. Weglarz, eds., Kluwer, 2003.

7. Ewa Deelman et al., "Pegasus: Mapping Scientific Workflows onto the Grid," to be published in *Proc. 2nd European Across Grids Conf.*, 2004.

8. E. Deelman et al., "Mapping Abstract Complex Workflows onto Grid Environments," *J. Grid Computing*, vol. 1, no. 1, 2003, pp. 25–39.

9. J. Blythe et al., "The Role of Planning in Grid Computing," *Proc. 13th Int'l Conf. Automated Planning and Scheduling* (ICAPS 2003), AAAI Press, 2003, pp. 153–163.

10. J. Blythe et al., "Transparent Grid Computing: A Knowledge-Based Approach," *Proc. 15th Innovative Applications of Artificial Intelligence Conf.* (IAAI), AAAI Press, 2003, pp. 57–64.

11. J. Frey et al., "Condor-G: A Computation Management Agent for Multi-institutional Grids," *Cluster Computing*, vol. 5, 2002, pp. 237–246.

12. I. Foster, C Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputer Applications*, vol. 15, no. 2, 2001; www.globus.org/research/papers/anatomy.pdf.

13. O. Lassila and R.R. Swick, *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, World Wide Web Consortium, 22 Feb. 1999; www.w3.org/TR/1999/REC-rdf-syntax-19990222.

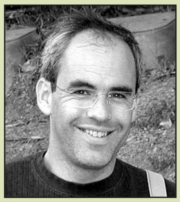
14. E. Christensen et al., *Web Services Description Language (WSDL) 1.1*, World Wide Web Consortium, 15 Mar. 2001; www.w3.org/TR/wsdl.



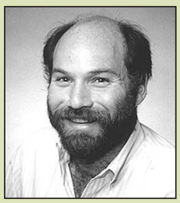
## The Authors

**Yolanda Gil**'s bio and photo can be found on page 25.

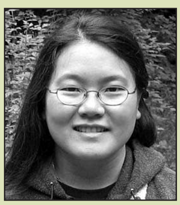
**Ewa Deelman** is a research team leader at the Center for Grid Technologies at the University of Southern California's Information Sciences Institute and an assistant research professor at the USC Computer Science Department. Her research interests include the design and exploration of collaborative scientific environments based on grid technologies, with particular emphasis on workflow management as well as the management of large amounts of data and metadata. At ISI she is part of the Globus project, which designs and implements Grid middleware. She received her PhD from Rensselaer Polytechnic Institute in Computer Science in 1997 in the area of parallel discrete event simulation. Contact her at USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292; deelman@isi.edu.



**Jim Blythe** is a computer scientist in USC's Information Sciences Institute. His research interests include planning, grid workflows, and knowledge acquisition for problem solving and planning applications. He received his PhD in computer science from Carnegie Mellon University. Contact him at the ISI/USC, 4676 Admiralty Way, Marina del Rey, CA 90292-6696; blythe@isi.edu.



**Carl Kesselman** is a fellow in USC's Information Sciences Institute. He is the director of the Center for Grid Technologies at ISI, and a research associate professor of computer science at USC. He received his PhD in computer science from the University of California, Los Angeles. Together with Ian Foster, he cofounded the Globus Project, a leading Grid research project that has developed the Globus Toolkit, the defacto standard for Grid computing. Contact him at USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292; carl@isi.edu.



**Hongsuda Tangmunarunkit** is a computer scientist in the Center for Grid Technologies at USC's Information Sciences Institute. Her research interests include grid computing, networking, and distributed systems. She received her MS in computer science from USC. She is a member of the ACM. Contact her at USC/ISI, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292; hongsuda@isi.edu; www.isi.edu/~hongsuda.