SERVICE-ORIENTED ENVIRONMENTS FOR DYNAMICALLY INTERACTING WITH MESOSCALE WEATHER

Within a decade after John von Neumann and colleagues conducted the first experimental weather forecast on the ENIAC computer in the late 1940s, numerical models of the atmosphere became the foundation of modern-day weather forecasting and one of the driving application areas in computer science. This article describes research that is enabling a major shift toward dynamically adaptive responses to rapidly changing environmental conditions.

Each year across the US, *mesoscale weather events*—flash floods, tornadoes, hail, strong winds, lightning, and localized winter storms—cause hundreds of deaths, routinely disrupt transportation and commerce, and lead to economic losses averaging more than US$13 billion.[1] Although mitigating the impacts of such events would yield enormous economic and societal benefits, research leading to that goal is hindered by rigid IT frameworks that can't accommodate the real-time, on-demand, dynamically adaptive needs of mesoscale weather research; its disparate, high-volume data sets and streams; or the tremendous computational demands of its numerical models and data-assimilation systems.

1521-9615/05/$20.00 © 2005 IEEE
Copublished by the IEEE CS and the AIP

KELVIN K. DROEGEMEIER, KEITH BREWSTER, MING XUE, AND DANIEL WEBER
University of Oklahoma
DENNIS GANNON AND BETH PLALE
Indiana University
DANIEL REED AND LAVANYA RAMAKRISHNAN
University of North Carolina
JAY ALAMEDA AND ROBERT WILHELMSON
US National Center for Supercomputing Applications
TOM BALTZER, BEN DOMENICO, DONALD MURRAY, MOHAN RAMAMURTHY, AND ANNE WILSON
University Corporation for Atmospheric Research
RICHARD CLARK AND SEPIDEH YALDA
Millersville University
SARA GRAVES, RAHUL RAMACHANDRAN, AND JOHN RUSHING
University of Alabama in Huntsville
EVERETTE JOSEPH AND VERNON MORRIS
Howard University

In response to the increasingly urgent need for a comprehensive national cyberinfrastructure in mesoscale meteorology—particularly one that can interoperate with those being developed in other relevant disciplines—the US National Science Foundation (NSF) funded a large information technology research (ITR) grant in 2003, known as Linked Environments for Atmospheric Discovery (LEAD). A multidisciplinary effort involving nine institutions and more than 100 scientists, students, and technical staff in meteorology, computer science, social science, and education, LEAD addresses the fundamental research challenges needed to create an integrated, scalable framework for adaptively analyzing and predicting the atmosphere.

LEAD's foundation is dynamic workflow orchestration and data management in a Web services framework. These capabilities provide for the use of analysis tools, forecast models, and data repositories, not in fixed configurations or as static recipients of data but rather as dynamically adaptive, on-demand systems that respond to weather as it evolves. Although mesoscale meteorology is the particular problem to which we've applied the LEAD concept, the methodologies and infrastructures we've developed are extensible to other domains such as medicine, ecology, oceanography, and biology.

In a companion article,[2] we describe the organization and cataloging of metadata as LEAD workflows generate them. Here, we present a more holistic view of LEAD and focus on the rationale behind and structure of its architecture. Now beginning its third of five years, LEAD is deploying several of its key services to the research and education communities. During the next 18 months, the project will focus on how streaming observations and analysis/simulation output can be used to modify workflows in real time.

Glossary of Acronyms

ADaM	Algorithm and Data Mining
ARPS	Advanced Regional Prediction System
BPEL	Business Process Execution Language
CASA	Center for Collaborative Adaptive Sensing of the Atmosphere
CDT	Central daylight time
CST	Central standard time
dBZ	Decibel Units of Radar Reflectivity
GRAM	Grid Resource Allocation and Management
HAPI	Health Application Programming Interface
ITR	Information technology research
LDM	Local data manager
LEAD	Linked Environments for Atmospheric Discovery
myLEAD	My Linked Environments for Atmospheric Discovery
NEXRAD	Next-generation radar
NSF	National Science Foundation
SOA	Service-oriented architecture
THREDDS	Thematic real-time environmental distributed data services
UTC	Coordinated universal time
WOORDS	Workflow orchestration for on-demand, real-time, dynamically adaptive systems
WRF	Weather research and forecasting model

The Case for Dynamic Adaptation

Those having experienced the devastation of a tornado or hurricane, the raging waters of a flash flood, or the paralyzing impacts of lake-effect snows understand that mesoscale weather develops rapidly, often with considerable uncertainty with regard to location. Such weather is also locally intense and frequently influenced by processes on both larger and smaller scales. Ironically, few of the technologies people use to observe the atmosphere, predict its evolution, and compute, transmit, or store information about it operate in a manner that accommodates mesoscale weather's dynamic behavior. Radars don't adaptively scan specific regions of thunderstorms; numerical models are run largely on fixed time schedules in fixed configurations; and cyberinfrastructure doesn't allow meteorological tools to run on-demand, change configuration in response to the weather, or provide the fault tolerance needed for rapid reconfiguration. As a result, today's weather technology is highly constrained and far from optimal when applied to any particular situation. To demonstrate, let's review examples in which different modes of adaptation within numerical models yield notable forecast improvements. We'll also look at the necessity of adaptive observations and cyberinfrastructure in creating a suitable environment for studying mesoscale weather.

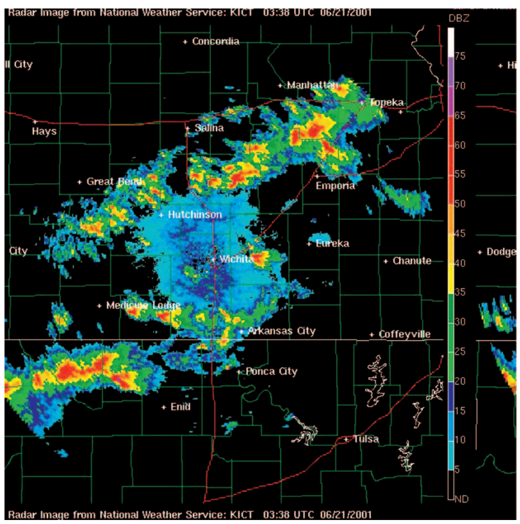


Figure 1. Radar reflectivity (proportional to precipitation intensity). Warmer colors indicate greater intensity on this radar image of storms over northeastern and central Kansas, on 21 June 2001. The radar is located in the center of the image at Wichita, Kansas.

Adaptation in Time

One of the most basic adaptive strategies in numerical weather prediction is rapid updating, in which the appearance of features in a given forecast suggests running successive forecasts more frequently. For mesoscale weather, which can appear suddenly and evolve rapidly, this capability is especially relevant, though not easily achieved owing to on-demand requirements for computing and data

Project Status

Given the LEAD service orchestration infrastructure's complexity and the number and diversity of services it offers, we're developing a series of LEAD prototypes that let us incrementally test and evaluate approaches and components within the controlled environment of the LEAD grid. All of this is a prelude to production deployment of LEAD's "science gateway" on the US National Science Foundation's TeraGrid, a networked collection of some of the largest computing and storage resources available in the US for open scientific research.

At this writing, the LEAD team has developed and integrated an operational portal with more than a dozen applications deployed as Web services and used in workflows. We've conducted experiments with the prototype framework atop the LEAD grid, and we estimate that 85 percent of the infrastructure is complete.

With regard to the LEAD architecture, all the cross-cutting services exist and have been deployed on one or more of the grid testbeds. The LEAD portal is operational but is expected to evolve as usage patterns and needs change. With the exception of the scheduler, replica locator, and the generic ingest service, all the resource-access services have been deployed. The workflow monitors and engines are operational, along with the virtual organization catalog and the THREDDS service. Initial versions of the application and configuration services exist and operate. Major services yet to be developed and deployed are the data stream services and application resource broker; both are critical components needed to move from static to dynamic workflows. Rather than build a new broker, we're evaluating the solutions being developed in several other TeraGrid and NSF ITR projects.

resources. To illustrate, Figure 1 shows reflectivity (equivalent to precipitation intensity) from the Wichita, Kansas, WSR-88D radar, also known as next-generation radar (NEXRAD), at 0336 coordinated universal time (UTC) or 10:36 p.m. central daylight time (CDT) on 21 June 2001. Clearly evident is a broken line of intense thunderstorms (bright red colors) oriented northeast–southwest and extending from just southwest of Topeka, Kansas, to south of Great Bend, Kansas. A second area of storms is present in northern Oklahoma.

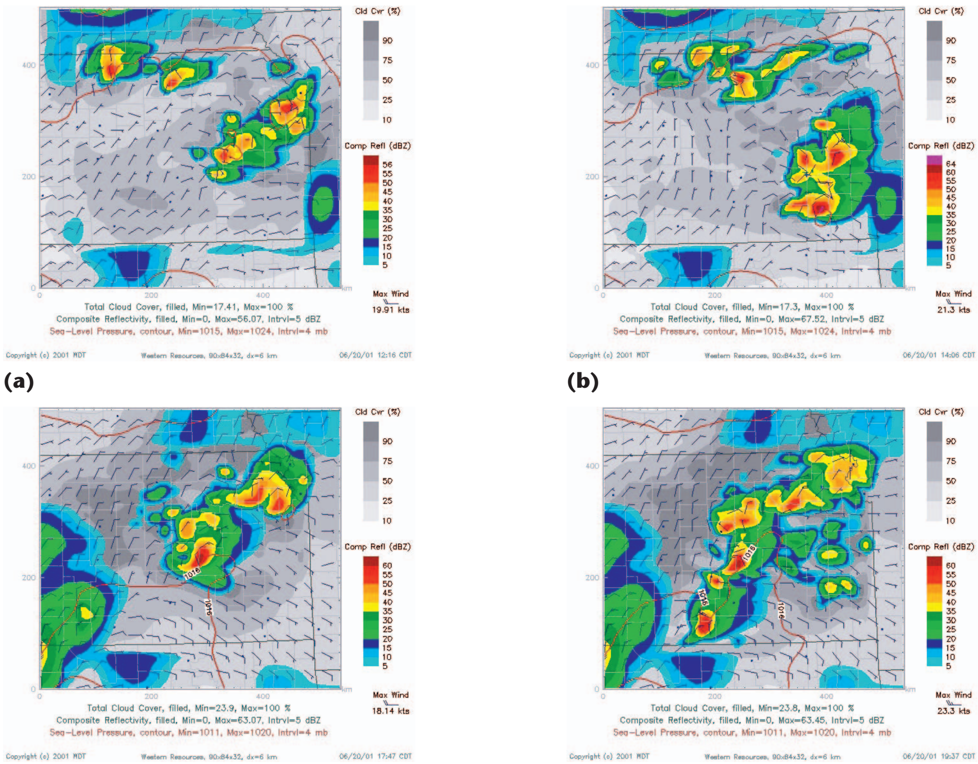
Just after noon that same day, an electric utility in Kansas used a customized version of a fine-scale computer prediction model called the Advanced Regional Prediction System (ARPS)[3] to generate the 11-hour forecast shown in Figure 2a. The utility initiated the forecast at 11 a.m. CDT and extended it through 10 p.m. CDT, or approximately 38 minutes prior to the radar image in Figure 1. The forecast depicts an area of thunderstorms having roughly the same alignment as what eventually developed, but before mobilizing repair crews to deal with possible power outages, the utility modified the model's execution schedule and ran a rapid update cycle, producing forecasts every two hours (Figures 2b through 2d). Although the nine-hour forecast (Figure 2b) produced a noticeably different solution from that initiated two hours earlier (Figure 2a), subsequent forecasts began to "lock onto" a consistent solution as the time of interest (10 p.m. CDT) approached, giving the utility confidence in the forecast and sufficient lead time to mobilize a response. By adapting the model to the weather and to its needs, the power utility took control of a potentially costly situation and mitigated loss.

Adaptation in Space

In addition to increasing forecast frequency as a means for obtaining more accurate solutions, models can adapt via the use of nested grids. This computational modality is quite common across a wide range of fluid dynamics applications, and researchers have automated it so that the grid mesh responds dynamically to changes in the flow using both structured[4] and unstructured[5] approaches. Such grid refinement is motivated by the desire to capture increasingly fine-scale features—particularly individual thunderstorms—along with the larger-scale environments in which they are embedded.

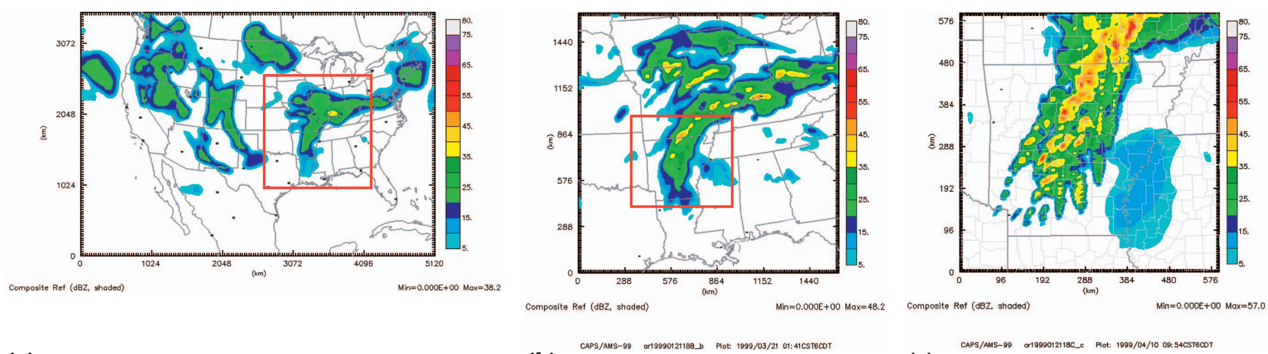
Figure 3a shows a 12-hour radar reflectivity forecast from the ARPS, valid at 0000 UTC or 6:00 p.m. central standard time (CST) on Friday, 29 January 1999, using a horizontal grid spacing of 32 km.[6] In reality, the northeast–southwest-oriented region of precipitation in Arkansas, which exhibits little fine-scale structure in the model due to the coarse grid, contained multiple lines of tornadic thunderstorms (see Figure 4).

In an attempt to capture more detail, a nested grid using 9-km horizontal spacing (the red box in Figure 3a) was spawned over a region of intense weather and yielded the six-hour forecast shown in Figure 3b. Some explicit evidence of intense thunderstorms emerges (yellow colors), although the 9-km grid is unable to resolve the most energetic elements of the flow—that is, individual updrafts and downdrafts. Spawning yet another grid at 3-km spacing (Figure



(a) (b) (c) (d)

Figure 2. Radar reflectivity forecasts. From the Advanced Regional Prediction System model on 20 June 2001, warmer colors indicate greater precipitation intensity at (a) the 11-hour forecast, (b) the nine-hour forecast, (c) the five-hour forecast, and (d) the three-hour forecast.



(a) (b) (c)

Figure 3. Radar reflectivity forecasts. (a) This 12-hour forecast on 22 January 1999 uses 32-km horizontal grid spacing; (b) a six-hour nested grid forecast using 9-km horizontal grid spacing shows some evidence of thunderstorms, but is still unable to capture individual cells. (c) A six-hour nested grid forecast using 3-km horizontal grid spacing over the domain shows remarkable agreement with observations in mode (broken lines of individual cells), orientation, and motion. The red boxes in panels (a) and (b) show the nested domain locations in (b) and (c), respectively.

3c), indicated by the red box in Figure 3b, yields a forecast that captures the multiple line structure, overall orientation, and generally correct movement of the storms (compare with Figure 4). Upon closer inspection, however, we see that the 3-km forecast does differ from observations in important ways (for example, the lack of storms in the "boot heel" of Missouri). Nevertheless, the ability to spatially adapt

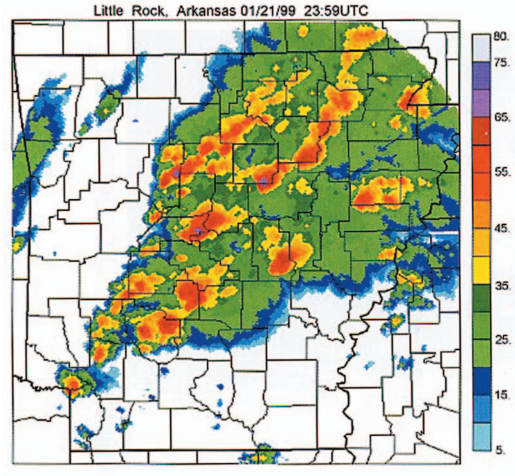


Figure 4. Multiple tornadic storms. Image from 21 January 1999 over Arkansas from multiple radars with their data objectively analyzed to a regular grid.[6]

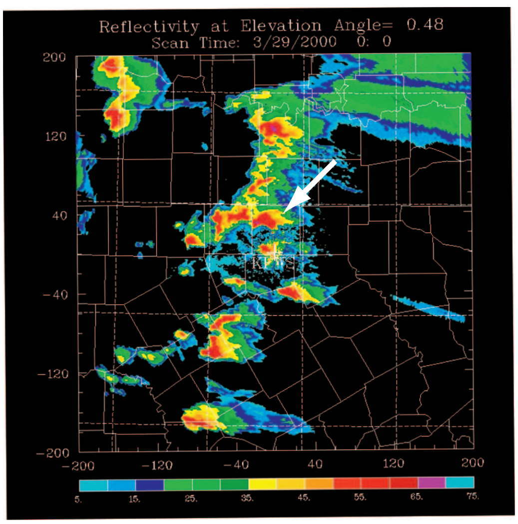


Figure 5. Observed storms. For a thunderstorm complex on 29 March 2000 over north central Texas, the white arrow shows the supercell that produced a major tornado in the Fort Worth metropolitan area.

the grid mesh (in this case, manually) clearly provides a positive impact by capturing individual thunderstorms that were absent at coarser grid spacings. Note, however, that such adaptation might not be warranted or desirable in all cases, nor may the requisite computational resources be available. By mining the model solution or observations for specified characteristics, LEAD provides the framework for intelligently, and automatically, generating nested domains in a grid context and providing quality of service estimates across the architecture.

Ensemble Forecasting

Comparing Figures 1 and 2, it's clear that any given forecast can contain considerable uncertainty, in large part because we never know the atmosphere's true state (due to incomplete sampling, observation errors, and so on). Consequently, a particular forecast's initial condition represents only one of numerous possibilities—that is, a single member of a probability distribution of physically plausible states. Because insufficient computational power exists to predict the full probability distribution's evolution (known as stochastic–dynamic forecasting),[7] meteorologists sample several states and produce numerous forecasts instead of making just one. This *ensemble methodology*—the creation of multiple concurrently valid forecasts from slightly different initial conditions, different models, the same model initialized at different times, or via the use of different physics options within the same or multiple models—has become the cornerstone of medium-range (six to 10 days) operational global numerical weather prediction;[8] in fact, it's even being extended to individual storms.[9] Of course, ensemble forecasting greatly increases the required computational resources and thus might be desirable only in certain situations, as dictated by the weather or a provisional forecast's outcome—thus, the need for intelligent, automated adaptation.

To illustrate the power of ensemble forecasting, Figure 5 shows radar reflectivity at 6 p.m. CST on 29 March 2000 over north central Texas; the figure is similar in content to Figure 1, except that it's from multiple radar data objectively analyzed to a regular grid. Clearly evident is a north–south-oriented line of intense thunderstorms, which ultimately produced multiple tornadoes, one of which passed through the Fort Worth, Texas, metropolitan area (white arrow), causing three deaths and nearly US$500 million dollars in damage.[10]

Researchers studying this case post facto initialized a five-member ensemble of forecasts at 2300 UTC on 29 March 2000. The control forecast in figure 6a captures the overall structure and motion of the storms in northern Texas, but it fails to predict the extension of the system further south. The other four ensemble members, initialized from slightly different (though equally probable) states and valid at the same time (see Figures 6b through 6e), exhibit considerable variability, with members 1 and 2 placing an extensive area of spurious storms in the southeastern part of the domain. Member 3 is different from all other forecasts, as well as reality (see Figure 5); if it were the only forecast avail-

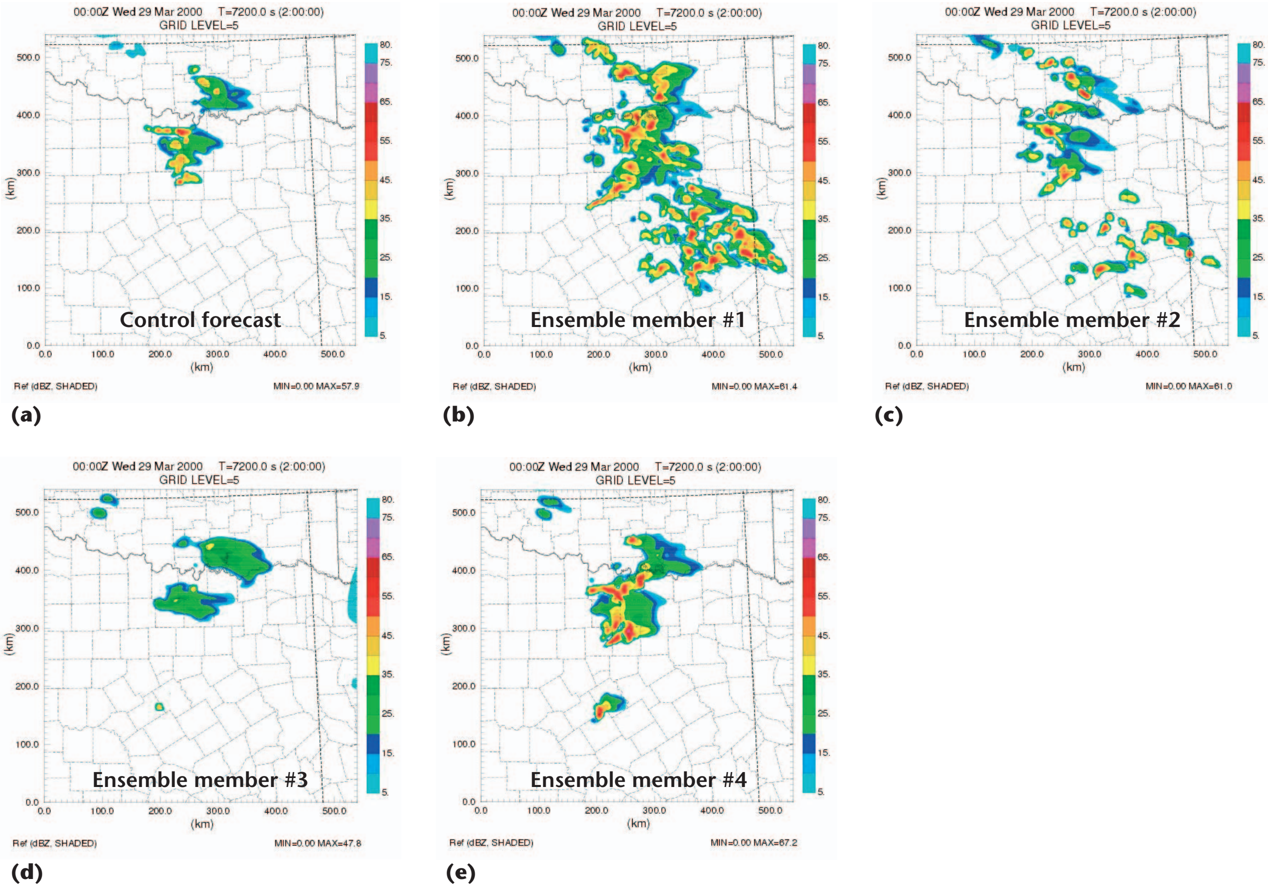


Figure 6. Five-member ensemble forecasting. In (a) a control forecast on 29 March 2000, warmer colors indicate greater precipitation intensity during a two-hour run; each of (b) – (e) the four ensemble members is created from a slightly different but physically plausible initial condition, yet the variation among predicted aerial coverage, intensity, and placement is significant.

able, the guidance would obviously be quite poor.

The practical value of ensemble forecasting lies in the ability to quantify forecast uncertainty and emphasize intense local events through the use of probabilities. Figure 7 shows the probability of radar reflectivity exceeding a value of 35 decibel units of radar reflectivity (dBZ), or heavy precipitation. This calculation simply involves determining—at each grid point—how many forecasts meet this criterion and then dividing by the total number of forecasts. Note how the ensemble de-emphasizes the spurious storms in the southeastern part of the domain and highlights the region in which all forecasts agreed—near Fort Worth—where all the tornadic storms actually occurred.

The ability to initiate an ensemble of forecasts automatically and then determine the ensemble's size, and thus the computational and networking load, dynamically based on a control run's output represents a significant adaptation to both observations and model output and is one of LEAD's key goals.

Adaptive Observing Systems

The adaptive behavior illustrated in the previous subsection is confined to the operation of a numerical model. However, models are fed by observations, the instruments for which are typically deployed in spatially regular arrays that remain fixed in space, collect observations at prescribed intervals, and operate largely independently (and in the same mode), regardless of the type of weather present. A prime example is the NEXRAD Doppler weather radar network across the US: due to the radars' long range, Earth's curvature prevents them from sampling approximately 72 percent of the atmosphere below 1 km. Furthermore, the radars have only a few modes of operation and can't be tasked to focus on specific regions of the atmosphere at the expense of others.

In recent years, researchers have supplemented conventional observation strategies with adaptive or targeted observations in which sensors are deployed to specific areas where additional informa-

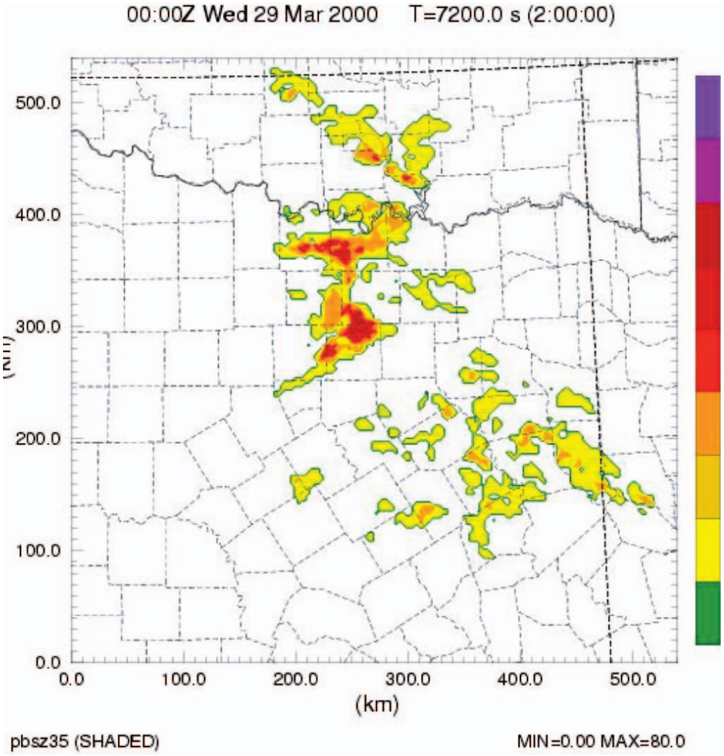


Figure 7. Probability of radar reflectivity exceeding 35 dBZ (moderate to heavy precipitation) based on the five-member ensemble forecast. The agreement among model solutions in the region of tornadic storms (upper center) yields the desired high probabilities.

tion is most likely improve forecast quality.[11] Examples include instruments dropped from aircraft and deployed on unmanned aerial vehicles. Although valuable, such strategies sample only a tiny fraction of the atmosphere and aren't suited to providing fine-scale, volumetric data in the interior of a thunderstorm, which is a domain of remote sensing exclusive to Doppler radar.

To address this problem, the NSF funded the Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere.[12] CASA seeks to revolutionize the sensing of the lowest 3 km of the atmosphere by developing inexpensive, low-cost, low-power Doppler radars that can be located on existing infrastructures such as cell-phone towers and buildings. Deployed in dense arrays, these radars are designed to collaborate with each other in real time, adaptively sensing multiple phenomena while simultaneously meeting multiple end-user needs. CASA represents the principal technology by which adaptive meteorological observations will be collected for use in LEAD; a four-radar CASA testbed in central Oklahoma will be operational in early 2006.

Adaptive Cyberinfrastructure

Achieving the types of adaptation we've described so far requires a flexible, fault-tolerant, dynamic cyberinfrastructure that can be rapidly and automatically reconfigured at the direction of remote sensors, models, and users. Moreover, high-performance computing and storage resources must be available with little advance notice, and data analysis and mining components must be able to detect faults, allow incremental processing, and estimate runtime and memory requirements based on evolving data properties. Owing to the highly perishable nature of the information being gathered and generated, network bandwidth must therefore be made available on demand to transfer large data sets and output files, and sufficient monitoring and quality of service guarantees must be available to support real-time experimentation and decision-making, especially in a classroom setting. In short, without an adaptive cyberinfrastructure, none of what we've just described is possible.

Managing the Forecast and Simulation Process

The potential value of dynamic adaptation is tempered by the reality of today's mesoscale meteorology research and education environments. Current weather tools such as data-ingest, quality-control, and analysis systems, as well as forecast models and post-processing environments, are enormously complex, even if used individually. They consist of sophisticated software developed over long time periods, contain numerous adjustable parameters and inputs, require users to deal with complex formats across a broad array of data types and sources, and often have limited transportability across computing architectures. When linked together and used with real data, the complexity increases dramatically.

Only a few academic institutions have the software infrastructure needed to automate the forecast process we described earlier. One is the University of Oklahoma and its Center for Analysis and Prediction of Storms, a graduated NSF Science and Technology Center that pioneered the science of thunderstorm prediction using numerical models. The software that manages its real-time forecast environment (http://caps.ou.edu/wx) consists of 50,000 lines of Perl code. This sophisticated and powerful software requires considerable programming expertise, is neither transportable nor scalable, and thus presents a huge barrier for graduate and undergraduate students as well as other users. The many other universities running experimental forecasts on a daily basis do so mostly in very simple configurations

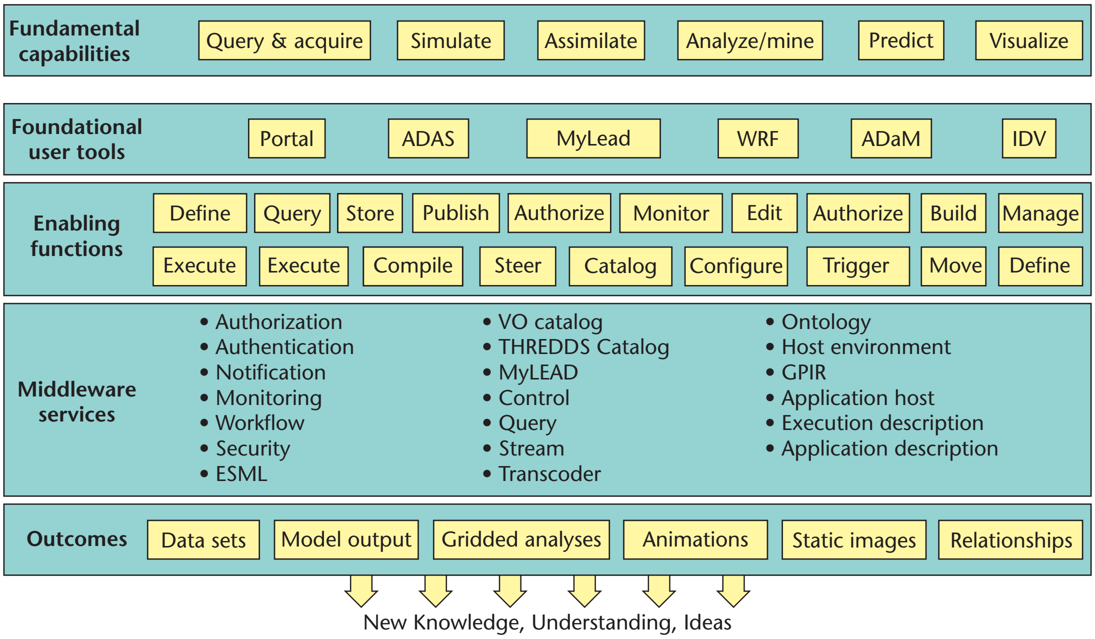


Figure 8. LEAD system. Fundamental capabilities familiar to meteorologists are shown in the top level, below which are the associated tools for enacting these capabilities and the middleware that links everything together. System-generated products appear at the bottom.

using local computing facilities and pregenerated analyses to which no new data are added. Thus, an important goal of LEAD is to lower the entry barrier for using complex weather technologies.

System Capabilities

LEAD's complex array of services, applications, interfaces, and local and remote computing, networking, and storage resources is assembled by users in workflows to study mesoscale weather as it evolves. Figure 8 shows the hierarchy of software required to create and orchestrate this suite of capabilities.

From the high-level view (the top of Figure 8), LEAD lets users query for and acquire information (for example, observational data sets, model results, the status of a resource or job, and so on), simulate and predict weather by using numerical atmospheric models, assimilate data (that is, combine observations under imposed dynamical constraints to create a 3D atmospheric state), and analyze, mine, and visualize data and model output. The outcomes of these operations (the bottom of Figure 8) include data sets, model output, gridded analyses, animations, static images, and a wide variety of relationships and other information. The fabric that links the user requirements with outcomes—namely, the extensive middleware, tool, and service capabilities—is LEAD's research domain.

We've provided several foundational tools within the LEAD environments (the second level in Figure 8), including

- a Web portal, the primary (though not exclusive) user entry point;
- the ARPS Data Assimilation System,[13] a sophisticated tool for data quality control and assimilation, including the preparation of initial conditions for simulations and forecasts;
- myLEAD,[14] a flexible metadata catalog service;
- the Weather Research and Forecast (WRF) model,[15] a next-generation atmospheric prediction and simulation model;
- ADaM (Algorithm Development and Mining),[16] a powerful suite of tools for mining observational data, assimilated data sets, and model output; and
- Integrated Data Viewer,[17] a widely used desktop application for visualizing a variety of multidimensional geophysical data.

As we'll describe later, these foundational tools and the resources that enable them are linked together in a service-oriented architecture (SOA).

System Concept

LEAD's conceptual underpinning is WOORDS, the workflow orchestration for on-demand, real-time, dynamically adaptive systems. As used in LEAD, WOORDS components have the following meaning:

- *Workflow orchestration* is the automation of a process, in whole or part, during which tasks or information are exchanged among system components to perform a specific action according to a set of procedural rules.
- *On demand* is the ability to perform an action immediately with or without prior planning or notification.
- *Real time* is the transmission or receipt of information about an event nearly simultaneously with its occurrence.
- *Dynamically adaptive* is the ability of a system, or any of its components, to respond automatically and in a coordinated manner to both internal and external influences.
- A *system* is a group of independent but interrelated elements that operate in a unified, holistic manner.

Another important notion is streaming data, which connotes information transported in a nearly time-continuous manner, often directly as input to services without first being written to files.

Although mesoscale meteorology and numerical weather prediction represent archetypal applications of WOORDS, the concept is far more general. The effective suppression of wild fires, for example, could depend on numerical simulations that incorporate evolving weather conditions, fuel availability, burn-line locations, and so on. Embedded sensors could measure roadway conditions and highway traffic flow to help reroute traffic in case of accidents. These examples show how using WOORDS as a general notion can benefit non-meteorology communities.

The LEAD System

LEAD consists of the following principal components (see Figure 9):

- The *user subsystem* comprises the LEAD portal—the principal mechanism by which users can access LEAD technologies—as well as the myLEAD personal workspace and the geo-reference graphical user interface.
- The *data subsystem* handles data and metadata, any numerical model output produced by operational or experimental models, and user-generated information.
- The *tools subsystem* consists of all meteorological and IT tools as well as interfaces for user-supplied tools.
- The *orchestration subsystem* provides the technologies that let users manage data flows and model execution streams, and create and mine output. It also provides linkages to other software and processes for continuous or on-demand applications.
- Located at six of the nine participating institutions (the University of Oklahoma, the University of Illinois at Urbana-Champaign, the University of Alabama in Huntsville, the UCAR Unidata Program, the University of North Carolina at Chapel Hill, and Indiana University), the distributed computing systems in the *LEAD grid* represent a distributed testbed for developing, integrating, and testing LEAD's components.

The LEAD system is instantiated as an SOA, which organizes an enterprise or system's key functions as a set of services. Workflows orchestrate the collections of service invocations and responses required to accomplish specific tasks. A Web service performs a specific operation, or a set of operations, based on requests from clients—for example, booking airline flights or looking up a friend's address. Web services conform to a family of standards, generally called "WS-*," that specify most aspects of the service's behavior. For example,

- the Web Service Definition Language (WSDL; www.w3c.org) specifies both how a service expects to receive requests and the type of responses it generates;
- WS-Addressing defines the way a client accesses a service and to what location the service should send responses; and
- WS-Security defines protocols for secure communication of messages from the client to the service and back.

SOAs are widely deployed in the commercial enterprise sector, and they form the foundation of many scientific "grid" technologies.

As Figure 10 shows, the LEAD SOA has five distinct yet highly interconnected layers. The bottom layer represents raw computation, application, and data resources distributed throughout the LEAD grid and elsewhere. The next level up holds the Web services that provide access to raw services such as those found in the Globus toolkit, as well as services for accessing weather data (Unidata's local data manager [LDM][18] and the Open-Source Project for a Network Data Access Protocol [www.opendap.org]) and data access services such as the replica location service[19] and the Open Grid Service Architecture Data Access and Integration (OGSA-DAI) service.[20]

The configuration and execution services in the middle layer, consisting of five elements, represent services invoked by LEAD workflows. The first is

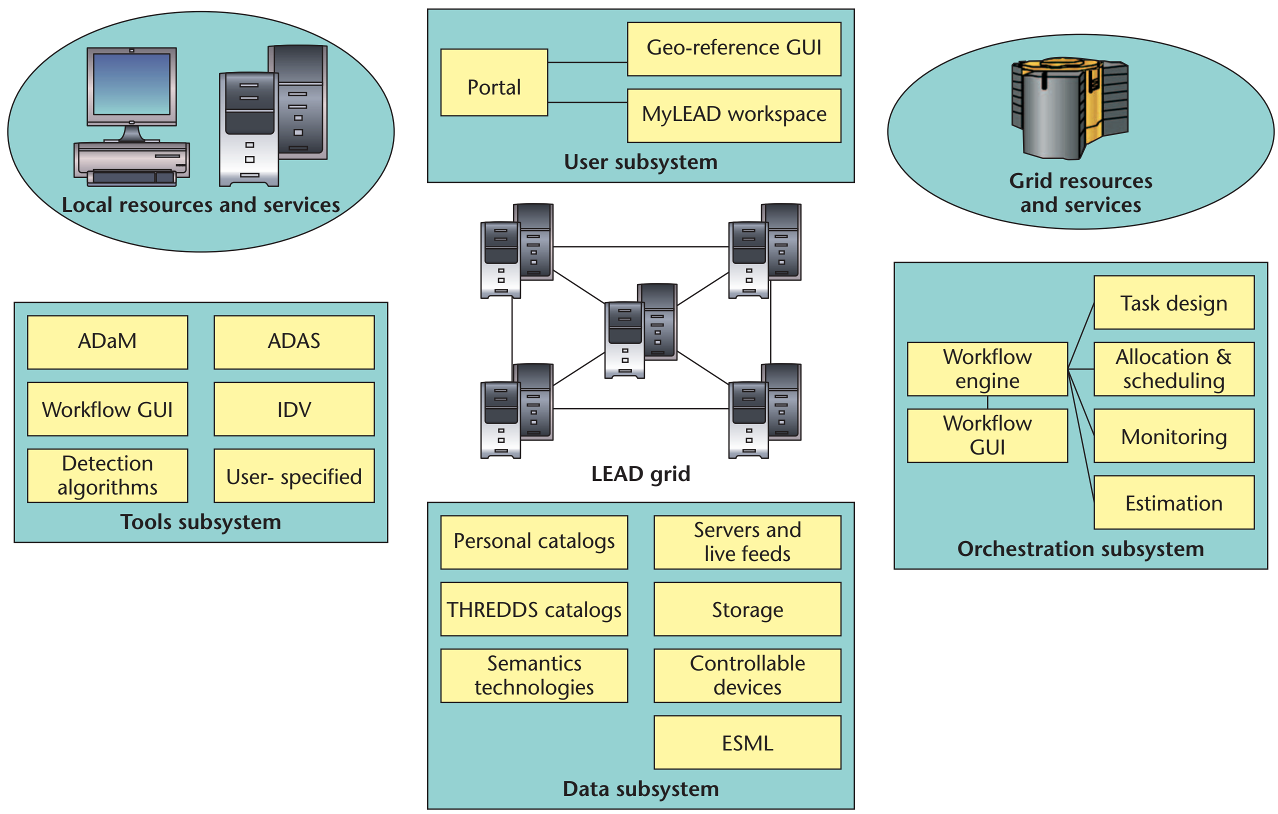


Figure 9. LEAD system framework. LEAD is composed of several interacting subsystems, with the LEAD grid representing a stable, secure environment for development and testing.

the application-oriented configuration service that manages the deployment and execution of real applications such as the WRF simulation model, the ARPS Data Assimilation System, and the ADaM tools. Related to it is the application resource broker, which matches the appropriate host for execution to each application task, based on the execution's time constraints. The workflow engine service, which drives experimental workflow instances, invokes both the configuration service and application resource broker. The fourth element, known as catalog services, represents the manner in which a user or application service discovers public-domain data products, or LEAD services, for use in computational experiments via a virtual organization catalog. This catalog obtains information about public data products by periodically crawling THREDDS (Thematic Real-time Environmental Distributed Data Services) catalogs,[21] which store pointers to a wide variety of data.

Finally, users require a host of data services to support rich query, access, and transformation operations on data products. An important goal behind LEAD is *access transparency*—facilitating user queries across all available heterogeneous data sources without adverse affects from different formats and naming schemes. Achieving any level of transparency requires at least minimal metadata for describing data products.

Metadata are essential for managing huge amounts of data generated by observing systems, models, and other meteorological resources. LEAD's XML schema—called the LEAD metadata schema—adheres to the US Federal Geographic Data Committee (FGDC)-defined standard for geospatial data. Specifically, we created an FGDC profile for LEAD by modifying the FGDC schema, restructuring it to handle the LEAD notion of resource collection, and adding other elements while still maintaining namespaces for separate LEAD catalogs. We released version 1.0 of the schema in June 2005.

A key Web service that maps higher-level atmospheric concepts to concrete terms used in data services is the LEAD ontology service. Decoder and interchange services, such as the Earth System Markup Language,[22] transform data from one form to another. Stream services manage live data

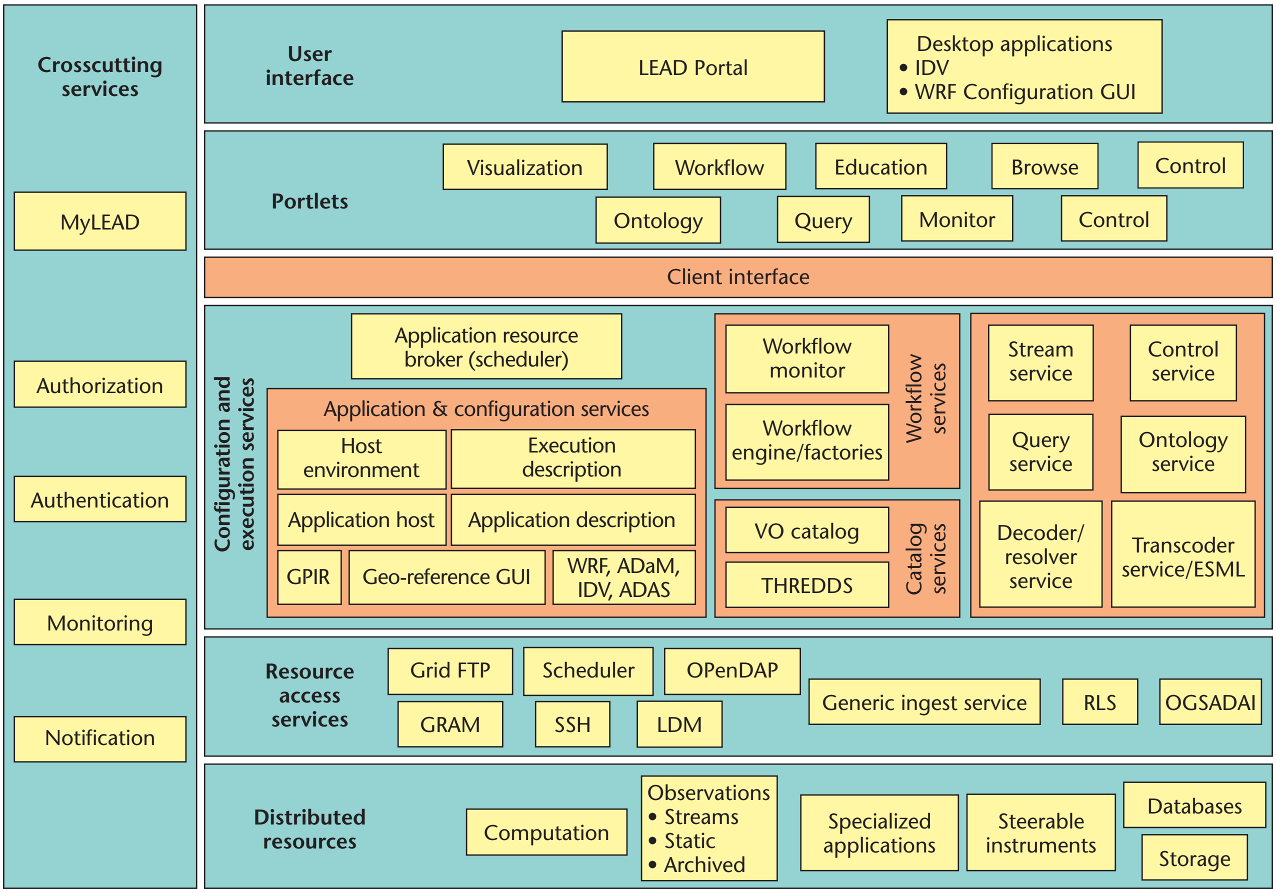


Figure 10. LEAD's service-oriented architecture. A wide variety of services and resources, grouped here according to general functionality, comprise the LEAD architecture, including services that cut across all aspects of the system.

streams such as those generated by the Nexrad Doppler radar network (with a per-radar bandwidth of approximately 14 Mbytes per five minutes, and 142 radars in the national network).

Users have access to several cross-cutting services within the LEAD SOA's layers. One such service—the notification service—lies at the heart of dynamic workflow orchestration. Each service can publish notifications, and any service or client can subscribe to receive them. This strategy is based on the WS-Eventing standard,[23] in which notifications signal task completion, job failure, or user commands. Another critical component is the monitoring service, which we'll discuss later. Monitoring provides, among other things, mechanisms to ensure that workflows are able to complete desired tasks by specified deadlines—an especially important issue in weather research.

Another vital cross-cutting service that ties multiple components together is the myLEAD user metadata catalog. As an experiment runs, it generates data stored on the LEAD grid and catalogued to the user's myLEAD metadata catalog (see Figure 11). Notification messages generated during the course of workflow execution are also written to the metadata and stored on each user's behalf. Users access metadata about data products via metadata-catalog-specific user interfaces built into the LEAD portal. Through these interfaces, they can then browse holdings, search for products on the basis of rich meteorological search criteria, publish and share products with broader groups, create snapshots of experiments for archiving, or upload text or notes to augment the experiment holdings.[24] Specialized services based on grid standards handle authentication and authorization.

The user interface to the system appears at the top level of the architecture in Figure 10. It consists of the LEAD Web portal and a collection of "service-aware" desktop tools. The portal is a container for user interfaces, called *portlets*, which provide access to individual services. When a user logs into the portal, his or her grid authentication and authorization credentials load automatically. Each portlet can use these identity certificates to access individual services on the users' behalf, thus allowing them to command the portal as a proxy for composing and executing workflows on back-end resources.

The LEAD Grid and Portal

We're currently deploying the LEAD SOA on the LEAD grid, which consists of a set of compute and storage resources (located at several of the LEAD institutions), ranging from single-CPU Linux systems with a few terabytes of local storage to large cluster-based systems and mass storage facilities capable of serving many petabytes of data. The LEAD grid is built on two systems: the Globus grid infrastructure framework[25] and the LDM. The LEAD SOA is layered on top of that.

The LEAD grid provides a distributed "clean room" environment within which to develop, integrate, and test the LEAD SOA. By having complete control over system and application software version control, LEAD can enforce compatibility requirements and avoid the numerous problems that can plague research and development efforts conducted in more open, unstable environments. We're currently developing strategies for migrating beyond this "safe sandbox" to other grids, such as the TeraGrid.

Most users access the LEAD grid via the LEAD portal (Figure 11), which also provides a gateway to the TeraGrid, a national infrastructure for computational science. The LEAD portal is based on the NSF National Middleware Initiative Open Grid Computing Environment Portal toolkit (www.ogce.org). This portal lets users load proxy identity certificates (based on Globus's GSI model) into the portal server, which in turn allows the portal server to interact with the LEAD grid on the user's behalf. The portal also provides the user with options for configuring experiments and launching them on the LEAD grid.

Workflows and Incremental Development

The LEAD system is a sequence of prototypes that serve to test and refine research concepts, engage end users, stimulate new ideas, and provide a mechanism for ensuring effective integration among multiple disciplines. The fundamental "building blocks" of these prototypes are a series of Web services (see Figure 12a) that also consist of services and can be used as standalone applications or as part of the overall LEAD environment. Users can combine the services, via an orchestration interface, in numerous ways to create a wide array of capabilities, adding other services as necessary or creating and saving workflows of services that can also be combined in new workflows to solve increasingly complex problems (see Figure 12b).

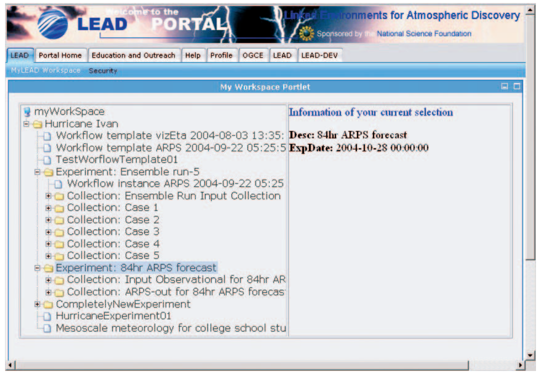


Figure 11. The LEAD portal. Upon logging in, the user is presented with a view of the myLEAD workspace, which is a private metadata catalog of results from the user's computational experiments.

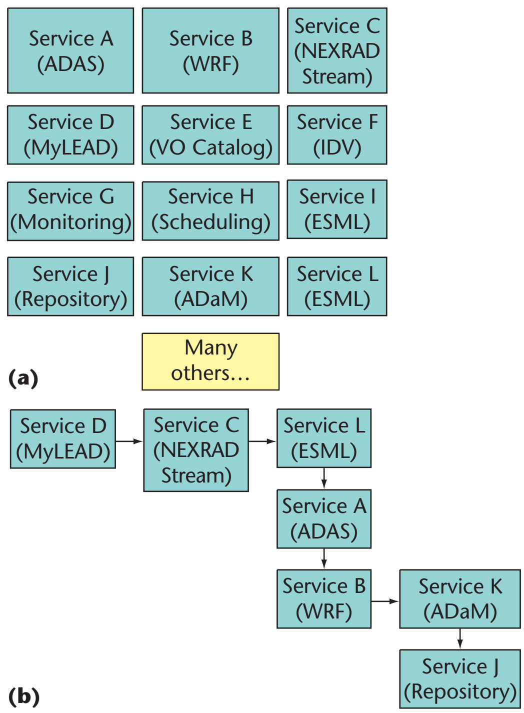


Figure 12. Web services and workflow engine. (a) LEAD is built on a series of Web services ranging from complex meteorological models to data streams and decoders. (b) By using the workflow engine to combine LEAD services, users can create a wide array of capabilities for research and education. Additional services can be added as needed.

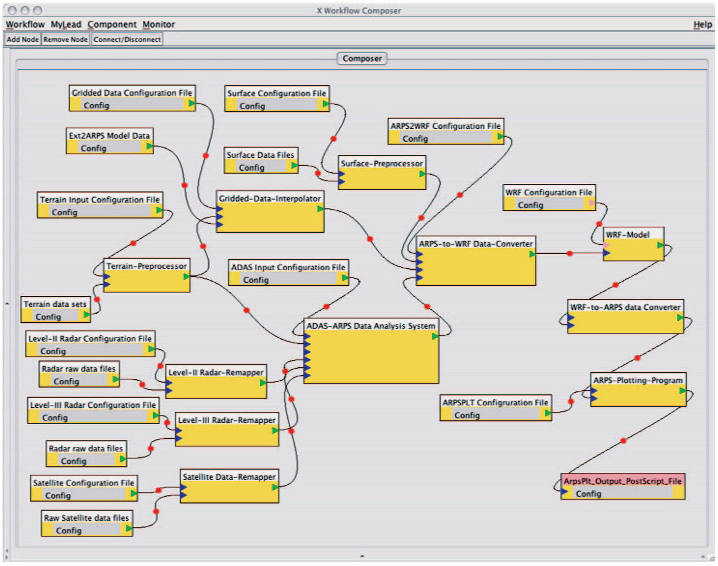


Figure 13. LEAD workflow. The composer shows the workflow for ingesting data, analyzing it, and then creating a numerical forecast.

Building Workflows

Because a complete experiment in LEAD might require the orchestration of a dozen or more application services, we describe it using a graphical tool that translates a "picture" of the workflow into a low-level workflow language. Figure 13 shows an actual LEAD workflow graph used to acquire and analyze atmospheric data, followed by the initiation of a numerical forecast. Each box in the graph is a Web service, a Web service-wrapped application, or an input parameter. Some of the application services merely stage files and start application tasks that run on compute clusters, such as those available through the TeraGrid. Other services function as brokers to provide access to compute resources by using an estimate of the computation work required and time constraints to select the resources needed to complete the task. The workflow instance, itself a Web service that runs in a workflow engine hosted on the LEAD grid, has a highly interactive relationship with the myLEAD personal metadata catalog.[2] The workflow shown in Figure 13, along with other workflows and the basic capabilities needed to create, edit, and execute them, are now running on the LEAD grid.

As shown schematically in Figure 14, LEAD is evolving three distinct yet related generations of workflow technology. In Generation 1, workflows are static—that is, all tasks to be performed, including their order of execution, data dependencies, and computational resources, must be determined prior to job launch and can't be changed until the job concludes. In Generation 2, the user can modify workflows during execution, or the workflow can modify itself in response to any number of conditions (such as loss of data, identification of new features in output or observations, or availability of computing resources). Furthermore, on-demand capabilities will become available in Generation 2, requiring sophisticated monitoring and performance-estimation resources, given the workflows' dynamic nature. Generation 3 will provide the capability for meteorological tools to interact mutually with adaptive remote sensors, most notably the CASA Doppler weather radars. Currently, LEAD supports static workflows executing services running across multiple LEAD grid nodes. Efforts in 2006 and 2007 will focus on automated, dynamic workflows.

Workflows in a Dynamically Adaptive Environment

A simulation's space and time adaptation requirements, the potential of adaptive instrumentation, and an adaptive cyberinfrastructure make LEAD workflow management unique. Traditional workflows involve static patterns that coordinate fixed sets of partners through carefully orchestrated activities. Humans are in the loop, but at prescribed points at which a specific decision or approval is required. In sharp contrast, LEAD workflows are event driven—for example, a data-mining agent could monitor a streaming radar data feed to detect a specific set of weather patterns. Once detected, the agent can notify the workflow engine. The workflow engine is a persistent Web service that can manage hundreds of workflow instances concurrently, so when it receives a message, it identifies the appropriate workflow instance (to which the message is addressed) and then advances the state of that workflow through its tasks until another message comes through. The workflow instance is then "suspended" and placed back into the database.

LEAD is building its workflows on the Business Process Execution Language (WS-BPEL),[26] a widely accepted industry standard that provides the relevant control constructs for modeling dynamic behavior. We've built a BPEL execution engine that works within the SOA security mechanisms defined by the LEAD grid. Because BPEL is completely service-oriented, it can query and respond to external services such as resource allocators and monitoring services. BPEL also has a well-defined "exception" model that allows a workflow to change course if an unexpected condition arises.

Unfortunately, BPEL isn't well suited to the casual scientific user. Consequently, we're providing

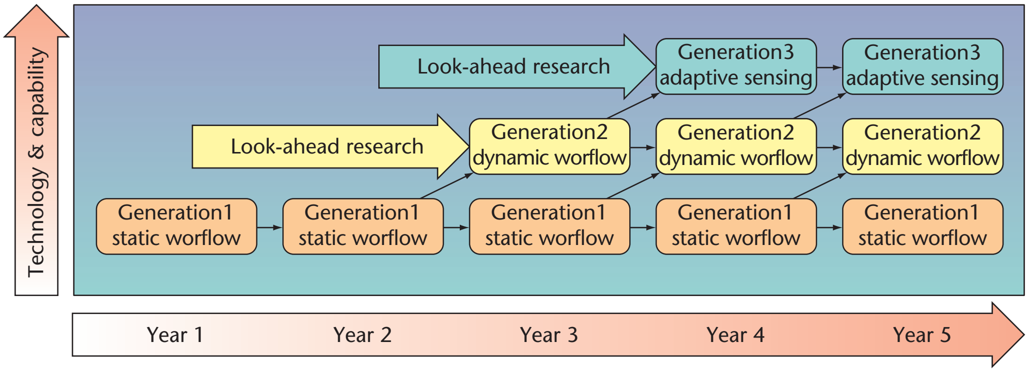


Figure 14. LEAD technology generations. We started with static workflows in the first generation, which served as the foundation for dynamic workflows beginning in year three, and will finally evolve into dynamically adaptive observations in years four and five.

a "drag and drop" composition tool that will let users intuitively, "graphically" construct workflows that are then translated into BPEL. Deciding the best way to design such workflows remains an open research problem.

Monitoring's Vital Role

Events that cause adaptive behavior can occur at any level in the system—in the atmosphere when a weather condition arises, a forecast-model analysis that results in directives to a local radar, the service layer in response to inefficiencies in an ongoing workflow execution, or at the hardware and system software layer in response to excessive computational or network loads. The complexity of the LEAD architecture's dynamic characteristics makes monitoring and understanding application resource behavior both critical and challenging. When reacting to crucial weather changes or changing resource availability, the LEAD system must proactively assess and detect system performance anomalies, enable recovery, and ensure continued operation. For the system to be responsive to simultaneously occurring high-priority events, it uses detection services to detect, sense, and monitor the environment.

Before a workflow can begin execution, it must be mapped to a set of physical resources that can meet expected performance and reliability guarantees. The quality of service provided to a given workflow depends on its criticality, workflow component behavior, and the underlying execution resources' changing capabilities. In addition, the execution resources' geographically distributed nature introduces new failure modes, so monitoring services must be able to monitor and predict possible resource losses. In turn, the LEAD system must be able to use knowledge of application needs and the current resource state, obtained from the monitoring services, to allocate new resources to the workflow. This allocation's goal is to enable timely execution of the workflow with the desired performance and reliability guarantees.

To support real-time monitoring of distributed LEAD workflows, we developed a new infrastructure that combines elements of the Autopilot distributed performance monitoring toolkit,[27] the SvPablo application performance toolkit,[28] a newly created Health Application Monitoring Interface (HAPI), and a workflow-annotation system that shows the state of executing workflow elements. Autopilot sensors execute as part of the monitoring services to collect performance data periodically. At each stage, the workflow engine publishes events about the workflow's progress (for example, the workflow started, LDM started, or WRF ended). A central control service subscribes to these events to monitor the workflow's progress, which enables a correlation between collected performance data and workflow progress. This central service can monitor progress during workflow execution or trigger scheduling/rescheduling decisions during the workflow's orchestration.

We're exploiting the SvPablo application instrumentation and performance tuning toolkit to capture data on the interaction of workflow components with system resources. By measuring the performance of computation-intensive components such as the WRF model, we can optimize component performance as well as improve workflow scheduling.

Figure 15 shows the LEAD monitoring system that's currently operational on the LEAD grid. The visual representation of the workflow progress extends the workflow composer and displays the sta-

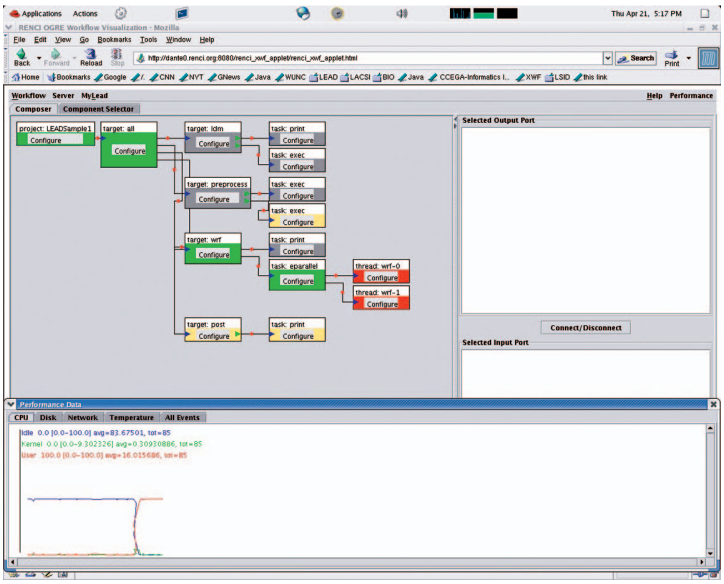


Figure 15. Workflow monitoring system. The LEAD monitoring system provides both graphical and textual information about the health and status of each workflow service and predicts hardware reliability by tracking CPU temperature and other variables.

tus of the workflow's tasks with a color-coded display. Different colors represent status, and color intensities represent the collective resource information's CPU load values (for example, yellow indicates the task hasn't started, and green represents the currently active branch of the workflow). Trends in performance and reliability on relevant resources are displayed below the workflow graph.

A reliability-monitoring toolkit complements the performance-monitoring infrastructure. It helps LEAD use reliability data to make scheduling decisions, anticipate likely failures, and take action before workflows are disrupted. The failure indicator infrastructure in LEAD is based on HAPI, an interface for discovery and use of health-related diagnostic information. By combining performance and failure models, the infrastructure can guarantee continued operation of weather forecasting and respond to additional resource requests based on weather phenomena.

Dynamically Adaptive Learning

In addition to creating a dynamically adaptive cyberinfrastructure for mesoscale meteorology, LEAD is evolving a consistent education concept known as the Dynamically Adaptive Learning Environment (DALE). In contrast to more conventional learning in which a student proceeds through a largely deterministic pathway toward understanding a particular concept, DALE places the student in an inquiry-based environment in which he or she can explore new ideas by creating entirely new pathways and tools—specifically, interacting with weather to explore new questions as they emerge in the student's mind.

Tools and services that offer student-centered capabilities are predicated on the existence of rich, dynamic models of student understanding.[29] Such models depict key ideas that learners should understand, common learner conceptions and misconceptions, and how these ideas change over time as student understanding becomes increasingly sophisticated.[30] DALE provides students with the ability to explore their own ideas in virtual concept spaces and is scalable and extensible to other disciplines.

The construction of the LEAD system and grid has been under way for two years, and we've learned many lessons. Some are common to most grid-oriented projects—for example, security is always more difficult to implement in a multi-institutional environment than expected. Moreover, it's impossible to avoid software being installed in different ways on different hosts even in a clean-room testbed built from dedicated resources, as is the case with the LEAD grid. Consequently, the construction of distributed applications with complete dependence on system-level coherence is virtually impossible. This fact reveals the distinct advantage of an SOA: interacting with service interfaces hides low-level system differences.

Finally, we've learned that maintaining large numbers of continuously running or persistent services is very difficult and places a substantial burden on system administrators. A viable solution is to build a system in which a small number of services (in our case, the crosscutting services and portal) are persistent, with the remainder instantiated on-demand by reliable core services. We anticipate learning many more lessons upon moving into the domain of dynamically adaptive workflows. CiSE

Acknowledgments

LEAD is a large information technology research grant funded by the US National Science Foundation under the following cooperative agreements: ATM-0331594 (University of Oklahoma), ATM-0331591 (Colorado State University), ATM-0331574 (Millersville University), ATM-0331480 (Indiana University), ATM-0331579 (University of Alabama in Huntsville), ATM03-31586 (Howard University), ATM-0331587 (University

Corporation for Atmospheric Research), and ATM-0331578 (University of Illinois at Urbana-Champaign, with a subcontract to the University of North Carolina).

References

1. R.A. Pielke and R. Carbone, "Weather Impacts, Forecasts, and Policy," *Bulletin Am. Meteorological Soc.*, vol. 83, 2002, pp. 393–403.
2. B. Plale et al., "Cooperating Services for Managing Data Driven Computational Experimentation," *Computing in Science & Eng.*, vol. 7, no. 5, 2005, pp. 34–43.
3. M. Xue, K.K. Droegemeier, and V. Wong, "The Advanced Regional Prediction System (ARPS): A Multiscale Nonhydrostatic Atmospheric Simulation and Prediction Model, Part I: Model Dynamics and Verification," *Meteorological and Atmospheric Physics*, vol. 75, 2000, pp. 161–193.
4. W.C. Skamarock and J.B. Klemp, "Adaptive Grid Refinement for Two-Dimensional and Three-Dimensional Nonhydrostatic Atmospheric Flow," *Monthly Weather Rev.*, vol. 121, 1993, pp. 788–804.
5. G. Dietachmayer and K. Droegemeier, "Application of Continuous Dynamic Grid Adaptation Techniques to Meteorological Modeling, Part I: Basic Formulation and Accuracy," *Monthly Weather Rev.*, vol. 120, 1992, pp. 1675–1706.
6. M. Xue et al., "The Advanced Regional Prediction System (ARPS): Storm-Scale Numerical Weather Prediction and Data Assimilation," *Meteorological and Atmospheric Physics*, vol. 82, 2003, pp. 139–170.
7. R.J. Fleming, "On Stochastic Dynamic Prediction I: The Energetics of Uncertainty and the Question of Closure," *Monthly Weather Rev.*, vol. 99, 1971, pp. 851–872.
8. E. Kalnay, *Atmospheric Modeling, Data Assimilation and Predictability*, Cambridge Press, 2003.
9. F. Kong, K.K. Droegemeier, and N.L. Levit, "Multiple Resolution Ensemble Forecast of an Observed Tornadic Thunderstorm System, Part I: Comparison of Coarse and Fine Grid Experiments," to be published in *Monthly Weather Rev.*, 2005.
10. *Storm Data*, Nat'l Climatic Data Center March 2000 Storm Data, vol. 42, no.3, 2000, p. 172.
11. R.E. Morss, K.A. Emanuel, and C. Snyder, "Idealized Adaptive Observation Strategies for Improving Numerical Weather Prediction," *J. Atmospheric Science*, vol. 58, 2001, pp. 210–232.
12. D.J. McLaughlin et al., "Distributed Collaborative Adaptive Sensing (DCAS) for Improved Detection, Understanding, and Prediction of Atmospheric Hazards," *Proc. 9th Symp. Integrated Observational and Assimilation Systems for the Atmosphere, Oceans, and Land Surfaces*, Am. Meteorological Soc., 2005.
13. K. Brewster, "Application of a Bratseth Analysis Scheme Including Doppler Radar Data," *Proc. 15th Conf. Weather Analysis and Forecasting*, Am. Meteorological Soc., 1996, pp. 92–95.
14. B. Plale et al., "User-Oriented Active Management of Scientific Data with myLEAD," *IEEE Internet Computing*, vol. 9, no. 1, 2005, pp. 27–34.
15. J. Michalakes et al., "Development of a Next-generation Regional Weather Research and Forecast Model," *Proc. 9th ECMWF Workshop on the Use of Parallel Processors in Meteorology*, Argonne Nat'l Lab., 2000; preprint ANL/MCS-P868-0101.
16. J. Rushing et al., "ADaM: A Data Mining Toolkit for Scientists and Engineers," to be published in *Computers & Geosciences*, 2005.
17. D. Murray et al., "The Integrated Data Viewer—A Web-Enabled Application for Scientific Analysis and Visualization, *Proc. 19th Conf. Integrated Information and Processing*, Am. Meteorological Soc., 2003.
18. R.K. Rew and G. Davis, "Distributed Data Capture and Processing in a Local Area Network," *Proc. 6th Int'l Conf. Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, *Am. Meteorological Soc.*, 1990, pp. 69–72.
19. S. Vazhkudai, S. Tuecke, and I. Foster, "Replica Selection in the Globus Data Grid," *IEEE/ACM Int'l Conf. Cluster Computing and the Grid* (CCGRID), IEEE CS Press, 2001, pp. 106–113.
20. M. Antonioletti et al., "Design and Implementation of Grid Database Services in OGSA-DAI," *Concurrency and Computation: Practice and Experience*, vol. 17, nos. 2–4, 2005, pp. 357–376.
21. E.R. Davis, and J. Caron, "THREDDS: A Geophysical Data/Metadata Framework," *Proc. 18th Int'l Conf. Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, Am. Meteorological Soc., 2002, pp. 52–53.
22. R. Ramachandran et al., "Earth Science Markup Language (ESML): A Solution for Scientific Data-Application Interoperability Problem," *Computers & Geosciences*, vol. 30, 2004, pp. 117–124.
23. D.L. Box et al., "Web Services Eventing (WS-Eventing)," Aug. 2004; http://ftpna2.bea.com/pub/downloads/WS-Eventing.pdf.
24. S. Lee et al., "Structure, Sharing, and Preservation of Scientific Experiment Data," to be published in *Proc. 3rd Int'l Workshop on Challenges of Large Applications in Distributed Environments* (CLADE), 2005.
25. I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *Int'l J. Supercomputer Applications*, vol. 11, no. 2, 1997, pp. 115–128.
26. T. Andrews et al., *Business Process Execution Language for Web Services*, v1.1, IBM Developers Library, May 2003.
27. D.A. Reed and C Mendes, "Intelligent Monitoring for Adaptation in Grid Applications," *Proc. IEEE*, vol. 93, no. 2, 2005, pp. 426–435.
28. L. DeRose, Y. Zhang, and D.A. Reed, "SvPablo: A Multi-Language Performance Analysis System," *Proc. 10th Int'l Conf. Computer Performance Evaluation, Modeling Techniques and Tools*, Springer-Verlag, 1998, pp. 352–355.
29. A.T. Corbett, K.R. Koedinger, and W.H. Hadley, "Cognitive Tutor: From the Research Classroom to All Classrooms," *Technology Enhanced Learning: Opportunities for Change*, P.S. Goodman, ed., Lawrence Erlbaum Assoc., 2001, pp. 235–263.
30. *Benchmarks for Science Literacy: New York, Project 2061*, Am. Assoc. for the Advancement of Science, Oxford Univ. Press, 1993.

Kelvin K. Droegemeier is Regents' Professor and Weathernews Chair of Meteorology, director of the Center for Analysis and Prediction of Storms at the Sasaki Institute, and associate vice president for research at the University of Oklahoma. His research interests are in numerical weather prediction, computational fluid dynamics, and high-performance computing. Droegemeier has a PhD in atmospheric science from the University of Illinois, Urbana-Champaign. Contact him at kkd@ou.edu.

Dennis Gannon is a professor in the Computer Science Department at Indiana University. His research interests include distributed systems, Grid computing, and building programming environments for scientific applications. Gannon has a PhD in computer science from the University of Illinois and a PhD in mathematics from the University of California, Davis. Contact him at gannon@cs.indiana.edu.

Daniel A. Reed is director of the Renaissance Computing Institute (RENCI), an interdisciplinary center spanning the University of North Carolina at Chapel Hill, Duke University, and North Carolina State University. He is also the vice-chancellor for information technology at the University of North Carolina at Chapel Hill, where he holds the Chancellor's Eminent Professorship. Contact him at Dan_Reed@unc.edu.

Beth Plale is an assistant professor in the computer science department at Indiana University. Her research interests include data management, grid computing, distributed systems, streaming data, and middleware. Plale has a PhD in computer science from the State University of New York Binghamton. Contact her at plale@cs.indiana.edu.

Jay Alameda leads the Middleware Division of the Integrated CyberService Directorate, US National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign. He has an MS in nuclear engineering from the University of Illinois, Urbana-Champaign, and a BS in chemical engineering from the University of Notre Dame. Contact him at jalameda@ncsa.uiuc.edu.

Tom Baltzer is a software engineer for the Unidata Program at the University Corporation for Atmospheric Research. He's also active in the Project Management Institute Mile High chapter, the Colorado Software Process Improvement Network, and the Boulder chapter of the ACM. Baltzer has a BS in computer science from the University of Colorado, Boulder. Contact him at tbaltzer@unidata.ucar.edu.

Keith Brewster is a senior research scientist with the Center for Analysis and Prediction of Storms at the University of Oklahoma. His interests are in the use of modern observing systems in high-resolution numerical weather prediction and hazards detection. Brewster has a PhD in meteorology from the University of Oklahoma. Contact him at kbrewster@ou.edu.

Richard D. Clark is the chair of the Department of Earth Sciences and professor of meteorology at Millersville University of Pennsylvania. His research interests include boundary layers and turbulence, air chemistry, and science education. Clark has a PhD in atmospheric science from the University of Wyoming. Contact him at Richard.Clark@millersville.edu.

Ben Domenico is deputy director of the Unidata Program Center at the University Corporation for Atmospheric Research. His research interests are in cyberinfrastructure for geoscientific data access, analysis, and visualization. Domenico has a PhD in astrophysics from the University of Colorado at Boulder. Contact him at ben@unidata.ucar.edu.

Sara J. Graves is a board of trustees university professor, professor of computer science, and director of the Information Technology and Systems Center at the University of Alabama in Huntsville. She is also the director of the Information Technology Research Center at the National Space Science and Technology Center. She has a PhD in computer science from the University of Alabama in Huntsville. Contact her at sgraves@itsc.uah.edu.

Everette Joseph is an associate professor in the Department of Physics and Astronomy at Howard University, codirector of the NOAA/Howard University Center for Atmospheric Sciences, and principal investigator on several research grants from NASA, NOAA, and NSF. He has a PhD in physics from the State University of New York at Albany. Contact him at ejoseph@howard.edu.

Don Murray is a meteorologist/software engineer at the Unidata Program Center in Boulder, Colorado. His interests include geoscience visualization, remote data access, glacial geology, and gardening. Murray has an MS in earth sciences from Montana State University. Contact him at dmurray@ucar.edu.

Rahul Ramachandran is a research scientist at the University of Alabama in Huntsville's Information Technology and Systems Center. His research interests include the design and development of scientific tools and algorithms, data mining, semantics, and ontology. Ramachandran has a PhD in atmospheric from the University of Alabama in Huntsville. Contact him at rramachandran@itsc.uah.edu.

Mohan Ramamurthy is the director of the University Corporation for Atmospheric Research's Unidata Program and is a scientist at the US National Center for Atmospheric Research. His technical interests include weather processes and prediction. Ramamurthy has a PhD in meteorology from the University of Oklahoma, Norman. Contact him at mohan@ucar.edu.

Lavanya Ramakrishnan is a research programmer at the Renaissance Computing Institute. Her research interests include grid and high-performance computing with a focus on monitoring and orchestration of application workflows and portal environments. Ramakrishnan has an MS in computer science from Indiana University, Bloomington. Contact her at lavanya@renci.org.

John A. Rushing is a senior research scientist at the University of Alabama in Huntsville. His research interests include pattern recognition, image processing, data mining, and artificial intelligence. Rushing has a PhD in computer science from the University of Alabama in Huntsville. Contact him at jrushing@itsc.uah.edu.

Dan Weber is a senior research scientist with the Center for Analysis and Prediction of Storms at the University of Oklahoma. His research interests include the development and use of numerical prediction models for investigating atmospheric convection, severe storms, and urban weather. Weber has a PhD in meteorology from the University of Oklahoma. Contact him at dweber@ou.edu.

Robert Wilhelmson is a professor of atmospheric sciences in the Department of Atmospheric Sciences and chief science officer and senior associate director of the US National Center for Supercomputing Applications, both at the University of Illinois at Urbana-Champaign. He has a PhD in computer science from the University of Illinois at Urbana-Champaign. Contact him at bw@ncsa.uiuc.edu.

Anne Wilson is a software engineer in the University Corporation for Atmospheric Research's Unidata Program, where she designs and builds systems that provide access to both near-real-time data and archival data. Wilson has a PhD in computer science from the University of Maryland, College Park. Contact her at anne@unidata.ucar.edu.

Ming Xue is an associate professor in the School of Meteorology and scientific director of the Center for Analysis and Prediction of Storms at the University of Oklahoma. His interests include the development and application of advanced numerical weather prediction and data-assimilation systems. Xue has a PhD in meteorology from the University of Reading, England. Contact him at mxue@ou.edu.

Sepideh Yalda is an associate professor of meteorology at Millersville University. Her research interests are in the areas of climate dynamics and regional climate change. Yalda has a PhD in meteorology from Saint Louis University. Contact her at Sepi.Yalda@millserville.edu.