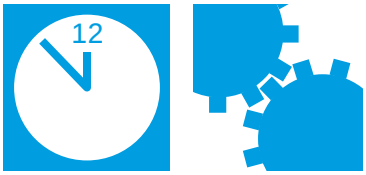


The MAGIC Project: From Vision to Reality

Barbara Fuller, Mitretek Systems
Ira Richer, Corporation for National Research Initiatives

Abstract

In the MAGIC project, three major components — an ATM internetwork, a distributed, network-based storage system, and a terrain visualization application — were designed, implemented, and integrated to create a testbed for demonstrating real-time, interactive exchange of data at high speeds among distributed resources. The testbed was developed as a system, with special consideration to how performance was affected by interactions among the components. This article presents an overview of the project, with emphasis on the challenges associated with implementing a complex distributed system, and with coordinating a multi-organization collaborative project that relied on distributed development. System-level design issues and performance measurements are described, as is a tool that was developed for analyzing performance and diagnosing problems in a distributed system. The management challenges that were encountered and some of the lessons learned during the course of the three-year project are discussed, and a brief summary of MAGIC-II, a recently initiated follow-on project, is given.

Gigabit-per-second networks offer the promise of a major advance in computing and communications: high-speed access to remote resources, including archives, time-critical data sources, and processing power. Over the past six years, there have been several efforts to develop gigabit networks and to demonstrate their utility, the most notable being the five testbeds that were supported by ARPA and National Science Foundation (NSF) funding: Aurora, BLANCA, CASA, Nectar, and VISTAnet [1]. Each of these testbeds comprised a mix of applications and networking technology, with some focusing more heavily on applications and others on networking. The groundbreaking work done in these testbeds had a significant impact on the development of high-speed networking technology and on the rapid progress in this area in the 1990s.

It became clear, however, that a new paradigm for application development was needed in order to realize the full benefits of gigabit networks. Specifically, network-based applications and their supporting resources. such as data servers, must be designed explicitly to operate effectively in a high-speed networking environment. For example, an interactive application working with remote storage devices must compensate for network delays. The MAGIC project, which is the subject of this article, is the first high-speed networking testbed that was implemented according to this paradigm. The major components of the testbed were considered to be interdependent parts of a system, and wherever possible they were designed to optimize end-to-end system performance rather than individual component performance.

The objective of the MAGIC (which stands for "Multidimensional Applications and Gigabit Internetwork Consortium") project was to build a testbed that could demonstrate real-time, interactive exchange of data at gigabit-per-second rates among multiple distributed resources. This objective was pursued through a multidisciplinary effort involving concurrent development and subsequent integration of three testbed components:

- An innovative terrain visualization application that requires massive amounts of remotely stored data
- A distributed image server system with performance sufficient to support the terrain visualization application
- A standards-based high-speed internetwork to link the computing resources required for real-time rendering of the terrain

The three-year project began in mid-1992 and involved the participation, support, and close cooperation of many diverse organizations from government, industry, and academia. These organizations had complementary skills and had the foresight to recognize the benefits of collaboration. The principal MAGIC research participants were:

- Earth Resources Observation System Data Center, U.S. Geological Survey (EDC)[1]
- Lawrence Berkeley National Laboratory, U.S. Department of Energy (LBNL)[1]
- Minnesota Supercomputer Center, Inc. (MSCI)[1]
- MITRE Corporation[1]
- Sprint
- SRI International (SRI)[1]

The work reported here was performed while the authors were with the MITRE Corp. in Bedford, MA, and was supported by the Advanced Research Project Agency (ARPA) under contract F19628-94-D-001.

[1]*These organizations were funded by ARPA.*

0890-8044/96/$05.00 © 1996 IEEE

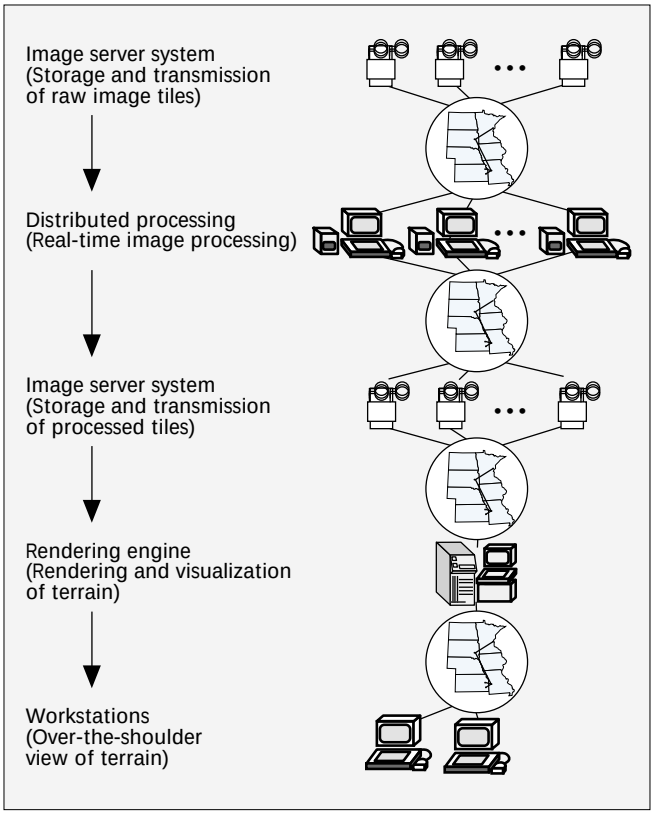


Figure 1. *Planned functionality of the MAGIC testbed.*

- University of Kansas (KU)[1]
- U S WEST Communications, Inc.

Other MAGIC participants that contributed equipment, facilities, and/or personnel to the effort were:

- Army High-Performance Computing Research Center (AHPCRC)
- Battle Command Battle Laboratory, U.S. Army Combined Arms Command (BCBL)
- Digital Equipment Corporation (DEC)
- Nortel, Inc./Bell Northern Research
- Southwestern Bell Telephone
- Splitrock Telecom

This article presents an overview of the MAGIC project with emphasis on the challenges associated with implementing a complex distributed system. Companion articles [2, 3] focus on a LAN/WAN gateway and a performance analysis tool that were developed for the MAGIC testbed. The article is organized as follows. The following section briefly describes the three major testbed components: the internetwork, the image server system, and the application. The third section discusses some of the system-level considerations that were addressed in designing these components, and the fourth section presents some high-level performance measurements. The fifth (affectionately entitled "Herding Cats") and sixth sections describe how this multi-organizational collaborative project was coordinated, and the technical and managerial lessons learned. Finally, the last section provides a brief summary of MAGIC-II, a follow-on project begun in early 1996.

Overview of the MAGIC Testbed

One of the primary goals of the MAGIC project was to create a testbed to demonstrate advanced capabilities that would not be possible without a very high-speed internetwork. MAGIC accomplished this goal by implementing an interactive terrain visualization application, TerraVision, that relies on a distributed image server system (ISS) to provide it with massive amounts of data in real time. The planned functionality of the MAGIC testbed is depicted in Fig. 1. Currently, TerraVision uses data processed off-line and stored on the ISS. In the future the application will be redesigned to enable real-time image processing as well as real-time terrain visualization (see the last section). Note that the workstations which house the application, the servers of the ISS, and the "over-the-shoulder" tool (see subsection entitled "The Terrain Visualization Application"), as well as those that will perform the on-line image processing, can reside anywhere on the network.

The MAGIC Internetwork

The MAGIC internetwork, depicted in Fig. 2, includes six high-speed local area networks (LANs) interconnected by a wide area network (WAN) backbone. The backbone, which spans a distance of approximately 600 miles, is based on synchronous optical network (SONET) technology and provides OC-48 (2.4 Gb/s) trunks, and OC-3 (155 Mb/s) and OC-12 (622 Mb/s) access ports. The LANs are based on asynchronous transfer mode (ATM) technology. Five of the LANs — those at BCBL in Fort Leavenworth, Kansas, EDC in Sioux Falls, South Dakota, MSCI in Minneapolis, Minnesota, Sprint in Overland Park, Kansas, and U S WEST in Minneapolis, Minnesota — use FORE Systems models ASX-100 and ASX-200 switches with OC-3c and 100 Mb/s TAXI interfaces. The ATM LAN at KU in Lawrence, Kansas, uses a DEC AN2 switch, a precursor to the DEC GigaSwitch/ATM, with OC-3c interfaces. The network uses permanent virtual circuits (PVCs) as well as switched virtual circuits (SVCs) based on both SPANS, a FORE Systems signaling protocol, and the ATM Forum User-Network Interface (UNI) 3.0 Q.2931 signaling stan-

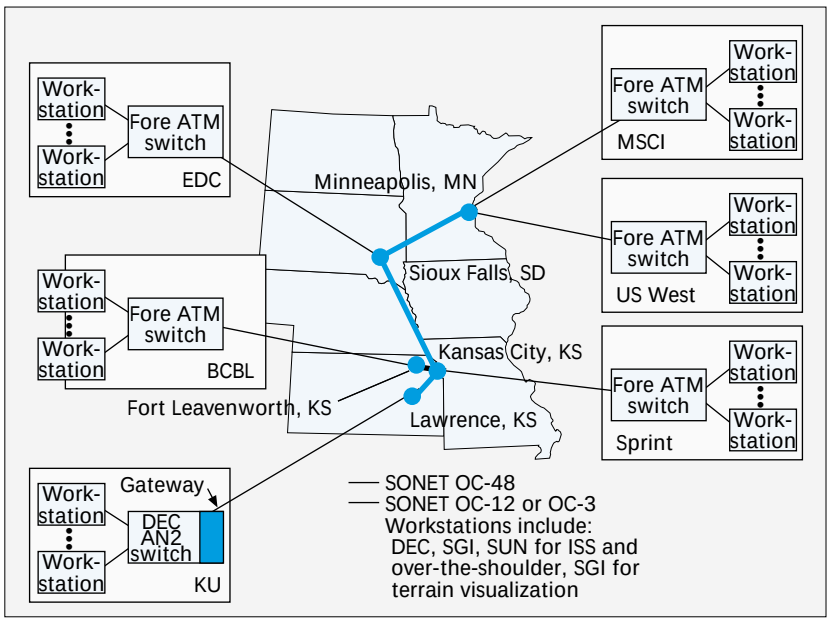


Figure 2. *Configuration of the MAGIC ATM internetwork.*

dard. The workstations at the MAGIC sites include models from DEC, SGI, and Sun. As part of MAGIC, an AN2/SONET gateway with an OC-12c interface was developed to link the AN2 LAN at KU to the MAGIC backbone [2].

In addition to implementing the internetwork, a variety of advanced networking technologies were developed and studied under MAGIC. A high-performance parallel interface (HIPPI)/ATM gateway was developed to interface an existing HIPPI network at MSCI to the MAGIC backbone. The gateway is an IP router rather than a network-layer device such as a broadband integrated services digital network (B-ISDN) terminal adapter, and was implemented in software on a high-performance workstation (an SGI Challenge). This architecture provides a programmable platform that can be modified for network research, and in the future can readily take advantage of more powerful workstation hardware. In addition, the platform is general-purpose; that is, it is capable of supporting multiple HIPPI interfaces as well as other interfaces such as fiber distributed data interface (FDDI).

Software was developed to enable UNIX hosts to communicate using Internet Protocol (IP) over an ATM network. This IP/ATM software currently runs on SPARCstations under Sun OS 4.1 and includes a device driver for the FORE SBA series of ATM adapters. It supports PVCs, SPANS, and UNI 3.0 signaling, as well as the "classical" IP and Address Resolution Protocol (ARP) over ATM model [4]. The software should be extensible to other UNIX operating systems, ATM interfaces, and IP/ATM address-resolution and routing strategies, and will facilitate research on issues associated with the integration of ATM networks into IP internets.

In order to enhance network throughput, flow-control schemes were evaluated and applied, and IP/ATM host parameters were tuned. Experiments showed that throughput close to the maximum theoretically possible could be attained on OC-3 links over long distances. To achieve high throughput, both the maximum transmission unit (MTU) and the Transmission Control Protocol (TCP) window must be large, and flow control must be used to ensure fairness and to avoid cell loss if there are interacting traffic patterns [5, 6].

The Terrain Visualization Application

TerraVision allows a user to view and navigate through (i.e., "fly over") a representation of a landscape created from aerial or satellite imagery [7]. The data used by TerraVision are derived from raw imagery and elevation information which have been preprocessed by a companion application known as TerraForm. TerraVision requires very large amounts of data in real time, transferred at both very bursty and high steady rates. Steady traffic occurs when a user moves smoothly through the terrain, whereas bursty traffic occurs when the user jumps ("teleports") to a new position. TerraVision is designed to use imagery data that are located remotely and supplied to the application as needed by means of a high-speed network. This design enables TerraVision to provide high-quality, interactive visualization of very large data sets in real time. TerraVision is of direct interest to a variety of organizations, including the Department of Defense. For example, the ability of a military officer to see a battlefield and to share a common view with others can be very effective for command and control.

Terrain visualization with TerraVision involves two activities: generating the digital data set required by the application, and rendering the image. MAGIC's approach to accomplishing these activities is described below. Enhancements to the application that provide additional features and capabilities are also described.

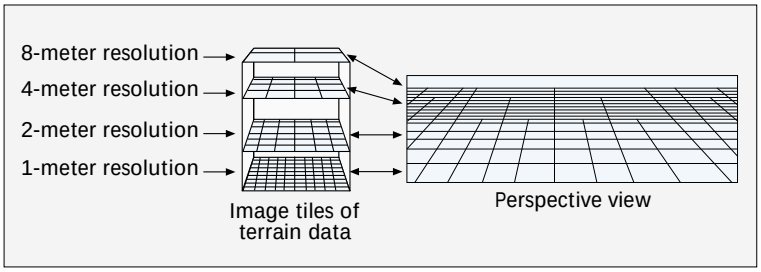


Figure 3. *Relationship between tile resolutions and perspective view. (Source: SRI International)*

Data Preparation — In order to render an image, TerraVision requires a digital description of the shape and appearance of the subject terrain. The shape of the terrain is represented by a two-dimensional grid of elevation values known as a *digital elevation model* (DEM). The appearance of the terrain is represented by a set of aerial images, known as *orthographic projection images* (ortho-images), that have been specially processed (i.e., ortho-rectified) to eliminate the effects of perspective distortion, and are in precise alignment with the DEM. To facilitate processing, distributed storage, and high-speed retrieval over a network, the DEM and images are divided into small fixed-size units known as *tiles*.

Low-resolution tiles are required for terrain that is distant from the viewpoint, whereas high-resolution tiles are required for close-in terrain. In addition, multiple resolutions are required to achieve perspective. These requirements are addressed by preparing a hierarchy of increasingly lower-resolution representations of the DEM and ortho-image tiles in which each level is at half the resolution of the previous level. The tiled, multiresolution hierarchy and the use of multiple resolutions to achieve perspective are shown in Fig. 3.

Rendering of the terrain on the screen is accomplished by combining the DEM and ortho-image tiles for the selected area at the appropriate resolution. As the user travels over the terrain, the DEM tiles and their corresponding ortho-image tiles are projected onto the screen using a perspective transform whose parameters are determined by factors such as the user's viewpoint and field of view. The mapping of a transformed ortho-image to its DEM and the rendering of that image are shown in Fig. 4.

The data set currently used in MAGIC covers a 1200 km^2 exercise area of the National Training Center at Fort Irwin, California, and is about 1 Gpixel in size. It is derived from aerial photographs obtained from the National Aerial Photography Program archives and DEM data obtained from the U.S. Geological Survey. The images are at approximately 1 m resolution (i.e., the spacing between pixels in the image corresponds to 1 m on the ground). The DEM data are at approximately 30 m resolution (i.e., elevation values in meters are at 30 m intervals).

Software for producing the ortho-images and creating the multiresolution hierarchy of DEM and ortho-image tiles was developed as part of the MAGIC effort. These processes were performed "off-line" on a Thinking Machines Corporation Connection Machine (CM-5) supercomputer owned by the AHPCRC and located at MSCI. The tiles were then stored on the distributed servers of the ISS and used by terrain visualization software residing on rendering engines at several locations.

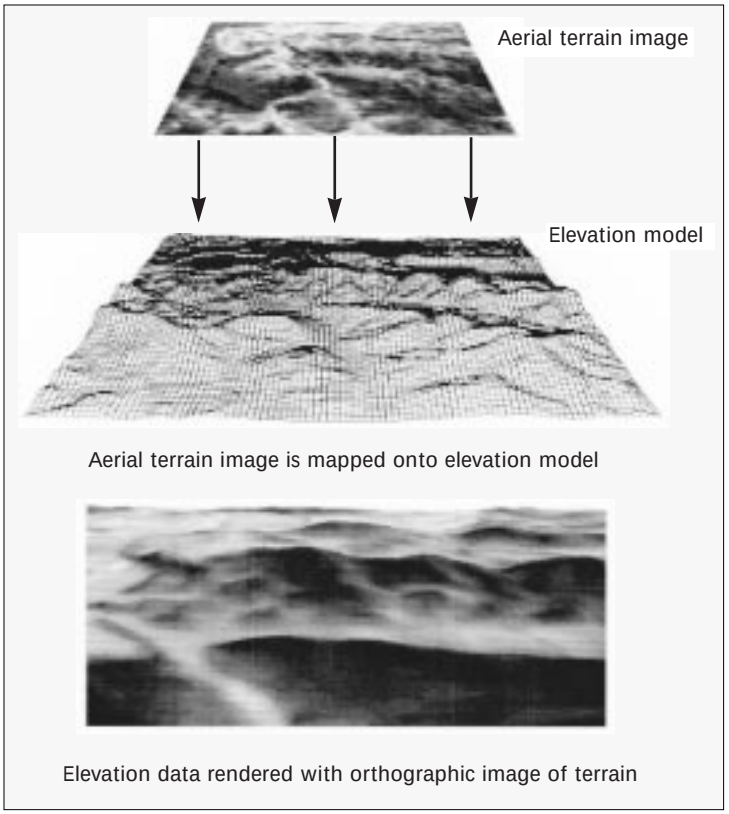


Figure 4. *Mapping an ortho-image onto its digital elevation model. (Source: SRI International)*

Image Rendering — TerraVision provides for two modes of visualization: two-dimensional (2-D) and three-dimensional (3-D). The 2-D mode allows the user to fly over the terrain, looking only straight down. The user controls the view by means of a 2-D input device such as a mouse. Since virtually no processing is required, the speed at which images are generated is limited by the throughput of the system comprising the ISS, the network, and the rendering engine.

In the 3-D mode, the user controls the visualization by means of an input device that allows six degrees of freedom in movement. The 3-D mode is computationally intensive, and satisfactory visualization requires both high frame rates (i.e., 15–30 frames/s) and low latencies (i.e., no more than 0.1 s between the time the user moves an input device and the time the new frame appears on the screen).

High frame rates are achieved by using a local very-high-speed rendering engine, an SGI Onyx, with a cache of tiles covering not only the area currently visible to the user, but also adjacent areas that are likely to be visible in the near future. A high-speed search algorithm is used to identify the tiles required to render a given view. For example, as noted above, perspective (i.e., 3-D) views require higher-resolution tiles in the foreground and lower-resolution tiles in the background. TerraVision requests the tiles from the ISS, places them in memory, and renders the view. Latency is minimized by separating image rendering from data input/output (I/O) so that the two activities can proceed simultaneously rather than sequentially (see the section entitled "Design Considerations").

Additional Features and Capabilities — TerraVision includes two additional features: superposition of fixed and mobile objects on the terrain, and registration of the user's viewpoint to a map. Both of these features are made possible by precisely aligning the DEM and imagery data with a world coordinate system as well as with each other.

A number of buildings and vehicles have been created and stored on the rendering engine for display as an overlay on the terrain. The locations of vehicles can be updated periodically by transferring vehicle location data, acquired with a global positioning system receiver, to the rendering engine for integration into the terrain visualization displays. Registration of the user's viewpoint to a map enables the user to specify the area he wishes to explore by pointing to it, and it aids the user in orienting himself.

In addition, an over-the-shoulder (OTS) tool was developed to allow a user at a remote workstation to view the terrain as it is rendered. The OTS tool is based on a client/server design and uses XWindow system calls. The user can view the entire image on the SGI screen at low resolution, and can also select a portion of the screen to view at higher resolution. The frame rate varies with the size and resolution of the viewed image, and with the throughput of the workstation.

The Image Server System

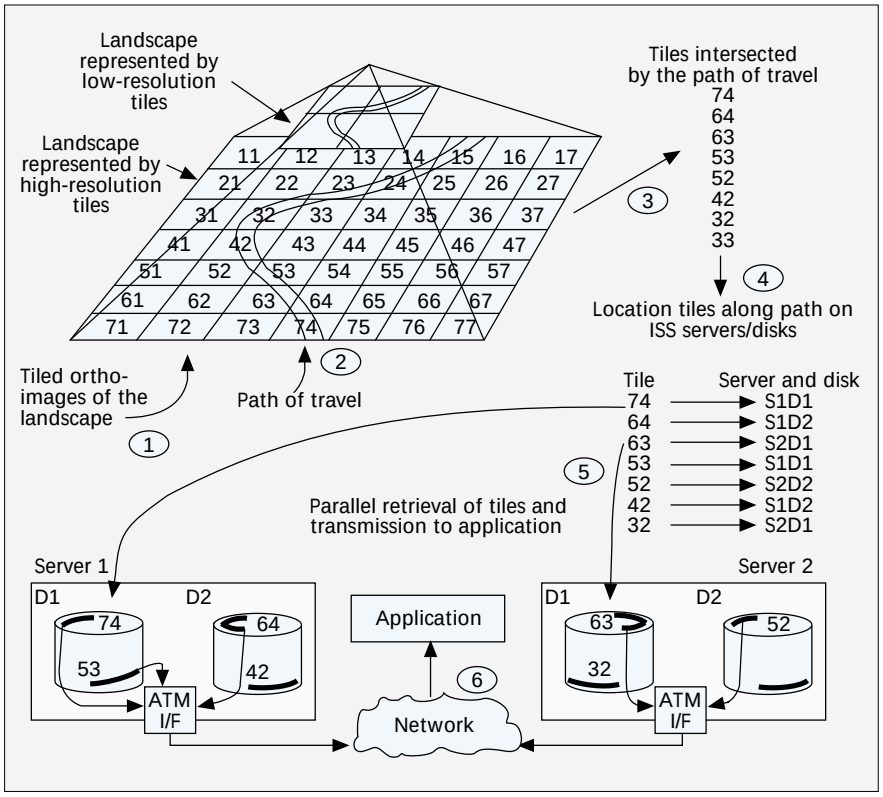
The ISS stores, organizes, and retrieves the processed imagery and elevation data required by TerraVision for interactive rendering of the terrain. The ISS consists of multiple coordinated workstation-based data servers that operate in parallel and are designed to be distributed around a WAN. This architecture compensates for the performance limitations of current disk technology. A single disk can deliver data at a rate that is about an order of magnitude slower than that needed to support a high-performance application such as TerraVision. By using multiple workstations with multiple disks and a high-speed network, the ISS can deliver data at an aggregate rate sufficient to enable real-time rendering of the terrain. In addition, this architecture permits location-independent access to databases, allows for system scalability, and is low in cost. Although redundant arrays of inexpensive disks (RAID) systems can deliver higher throughput than traditional disks, unlike the ISS they are implemented in hardware and, as such, do not support multiple data layout strategies; furthermore, they are relatively expensive. Such systems are therefore not appropriate for distributed environments with numerous data repositories serving a variety of applications.

The ISS, as currently used in MAGIC, comprises four or five UNIX workstations (including Sun SPARCstations, DEC Alphas, and SGI Indigos), each with four to six fast SCSI disks on two to three SCSI host adapters. Each server is also equipped with either a SONET or a TAXI network interface. The servers, operating in parallel, access the tiles and send them over the network, which delivers the aggregate stream to the host. This process is illustrated in Fig. 5. More details about the design and operation of the ISS can be found in [8].

Design Considerations

In MAGIC, the single most perspicuous criterion of successful operation is that the end user observes satisfactory performance of the interactive TerraVision application. When the user flies over the terrain, the displayed scene must flow smoothly, and when he teleports to an entirely

different location, the new scene must appear promptly. Obtaining such performance might be relatively straightforward if the terrain data were collocated with the rendering engine. However, one of the original premises underlying the MAGIC project is that the data set and the application are not collocated. There are several reasons for this, the most important being that the data set could be extremely large, so it might not be feasible to transfer it to the user's site. Moreover, experience has shown that in many cases the "owner" of a data set is also its "curator" and may be reluctant to distribute it, preferring instead to keep the data locally to simplify maintenance and updates. Finally, it was anticipated that future versions of the application might work with a mobile user and with fused data from multiple sources, and neither of these capabilities would be practical with local data. Therefore, since the data will not be local, the MAGIC components must be designed to compensate for possible delays and other degradations in the end-to-end operation of the system.



■ Figure 5. *Schematic representation of the operation of the ISS. (Source: Lawrence Berkeley National Laboratory)*

In order to understand system-level design issues, it is necessary to outline the sequence of events that occurs when the user moves the input device, causing a new scene to be generated. TerraVision first produces a list of new tiles required for the scene. This list is sent to an ISS master, which performs a name translation, mapping the logical address of each tile (the tile identifier) to its physical address (server/disk/location on disk). The master then sends each server an ordered list of the tiles it must retrieve. The server discards the previous list (even if it has not retrieved all the tiles on that list) and begins retrieving the tiles on the new list. Thus, the design for the system comprising TerraVision, the ISS, and the internetwork must address the following questions:

- How can TerraVision compensate for tiles it needs for the next image but have not yet been received?
- How often should TerraVision request tiles from the ISS?
- Where should the ISS master be located?
- How should tiles be distributed among the ISS disks?
- How can cell loss be minimized near the rendering site where the tile traffic becomes aggregated and congestion may occur?

Missing Tiles

Network congestion, an overload at an ISS server, or a component failure could result in the late arrival or loss of tiles that are requested by the application. Several mechanisms were implemented to deal with this problem. First, although the entire set of high-resolution tiles cannot be collocated with the application, it is certainly feasible to store a complete set of lower-resolution tiles. For example, if the entire data set comprises 1 Tbyte of high-resolution tiles, then all of the tiles that are five or more levels coarser would occupy less than 1.5 Mbyte, a readily affordable amount of local storage. If a tile with resolution at, say, level 3 is requested but not delivered in time for the image to be rendered, then, until the missing level-3 tile arrives, the locally available coarser tile from level 5 would be used in place of the 16 level-3 tiles. This substitution manifests itself by the affected portion of the rendered image appearing "fuzzy" for a brief period of time. Temporary substitution of low-resolution tiles for high-resolution tiles is particularly effective for teleporting because that operation requires a large number of new tiles, so it is more likely that one or more will be delayed.

Second, TerraVision attempts to predict the path the user will follow, requesting tiles that *might* soon be needed, and assigning one of three levels of priority to each tile requested. Priority-1 tiles are needed as soon as possible; the ISS retrieves and dispatches these first. This set of tiles is ordered by TerraVision, with the coarsest assigned the highest priority within the set. The reasons are:

- The rendering algorithm needs the coarse tiles before it needs the next-higher-resolution tiles.
- There are fewer tiles at the coarser resolutions, so it is less likely that they will be delayed.

The priority-2 tiles are those that the ISS should retrieve but should transmit only if there are no priority-1 tiles to be transmitted; that is, priority-2 tiles are put on a lower-priority transmit queue in the I/O buffer of each ISS server. (ATM switches would be allowed to drop the cells carrying these tiles.) Priority-3 tiles are those that should be retrieved and cached at the ISS server; these tiles are less likely to be needed by TerraVision. Note that there is a trade-off between "overpredicting" — requesting too many tiles — which would result in poor ISS performance and high network load, and "underpredicting," which would result in poor application performance.

Finally, a tile will continue to be included in TerraVision's request list if it is still needed and has not yet been delivered. Thus, tiles or tile requests that are dropped or otherwise "lost" in the network will likely be delivered in response to a subsequent request from the application.

Frequency of Requests

Another trade-off pertains to the frequency at which TerraVision sends its request list to the ISS. If the interval between requests is too large, then some tiles will not arrive when needed, resulting in a poor-quality display; in addition, the ISS will be idle and hence not used efficiently. On the other hand, if the interval is too short, then the request list might contain tiles that are currently in transit from servers to the application; this would result in poor ISS performance and redundant network traffic. For a typical MAGIC configuration, the interval between requests is currently set at 200 ms, a value that was found empirically to yield satisfactory performance. This value is based roughly on the measured latency of the ISS (about 100 ms) and on the estimated time required for a tile request to travel through the network from the TerraVision host to the ISS master and then to the most distant ISS server, plus the time for the tile itself to travel back to the host (perhaps a total of 50 ms). Additional measurements and analysis are needed to more precisely determine the appropriate request frequency as a function of the performance and location of system components and of network parameters.

For a typical MAGIC configuration, the interval between requests is currently set at 200 ms, a value that was found empirically to yield satisfactory performance.

Location of ISS Master

Since tile requests flow from TerraVision to the ISS master and thence to the servers themselves, the time for delivering the requests to the servers is minimized when the master is collocated with the TerraVision host. However, locating the master with the host is neither desirable nor practical for several reasons. The master is logically part of the ISS; therefore, its location should not be constrained by the application. Also, an ISS may be used with several applications concurrently, by multiple simultaneous users of a particular application, or by a user whose host may be unable to support any ISS functionality (e.g., a mobile user). Moreover, replication of the master would introduce problems associated with maintaining consistency among multiple masters when the ISS is in a read/write environment, as it would be when real-time data are being stored on the servers.

To first order, the delivery time of tile requests is limited by the time τ for a request to travel from TerraVision to the ISS server most distant from the TerraVision host. Hence, if the master is approximately on the path from the TerraVision host to that server, then τ will not be much greater than when the master and host are collocated. Furthermore, in the current MAGIC testbed, τ is much smaller than the sum of the disk latency and the network transit time. In other words, there is considerable freedom in choosing the location of the ISS master. Satisfactory system performance has been demonstrated, for example, with the TerraVision host in Kansas City, the ISS master in Sioux Falls, and servers in Minneapolis and Lawrence. Of course, this conclusion might change if faster servers reduce ISS latency considerably, or the geographic span of the network were substantially larger.

Distribution of Tiles on ISS Servers

The manner in which data are distributed among the servers determines the degree of parallelism and hence the aggregate throughput which can be obtained from the ISS. The data placement strategy depends on the application and is a function of data type and access patterns. For example, the retrieval pattern for a database of video clips would be quite different from that for a database of images. A strategy was developed for a terrain visualization type of application that minimizes the retrieval time for a set of tiles: the tiles assigned to a given disk are as far apart as possible in the terrain in order to maximize parallelism by minimizing the probability that tiles on a request list are on the same disk; and on each disk, tiles that are near each other in the terrain are placed as close as possible to minimize retrieval time. Although this was shown to be an optimal strategy for terrain path-following as in TerraVision [9], it was subsequently shown that ISS performance with random placement of tiles was only slightly worse. This was partly because tile retrieval time is much less than the latency in the ISS servers and network transit time, and is therefore not currently a significant factor in overall performance. Random placement is simpler to implement and is expected to be satisfactory for many other applications. However, as discussed for the location of the ISS master, this conclusion may have to be revisited if the performance or the geographic distribution of system components changes significantly.

Avoiding Cell Loss

When initially implemented, the MAGIC internetwork exhibited very low throughput in certain configurations. One cause of the low throughput was found to be mismatches between the burst rates of components in the communications path. Examples of such rate mismatches were:

- An OC-3 workstation interface transmitting cells at full rate across the network to a 100 Mb/s TAXI interface on another workstation
- Two or more OC-3 input ports at an ATM switch sending data to the same OC-3 output port

A mismatch, coupled with small buffers at the output ports of ATM switches, caused cells to be dropped, which in turn resulted in the retransmission of entire TCP packets, exacerbating the problem. In some cases the measured useful throughput was less than one percent of the capacity of the lower speed line.

Previously it was noted that in many cases a large MTU can increase throughput. However, once again there is a trade-off. As the MTU size is increased, the number of ATM cells needed to carry the MTU increases. The probability that one or more cells from the MTU will be dropped by the network therefore increases, which in turn increases the probability that the MTU will have to be retransmitted, thus possibly decreasing the effective throughput. Flow-control techniques together with large switch buffers and proper choice of protocol parameters did provide satisfactory performance. Nevertheless, the overriding conclusions are that the parameters of the entire end-to-end system, not just those of a single host or switch, must be tuned, each direction of the data path must be evaluated separately, and every component in each direction of the data path must be considered in the evaluation.

Performance

This section presents highlights of system-level performance issues and measurements of the MAGIC testbed. The input data rates that are needed to support the TerraVision application are calculated first, to provide the context for the subsequent discussion. Then the data rates that the network, the ISS, and the application host can actually support are described. Finally, a diagnostic tool that was developed to help analyze system performance is explained. More detailed information about all of the above topics can be found in [3, 5, 6, 10].

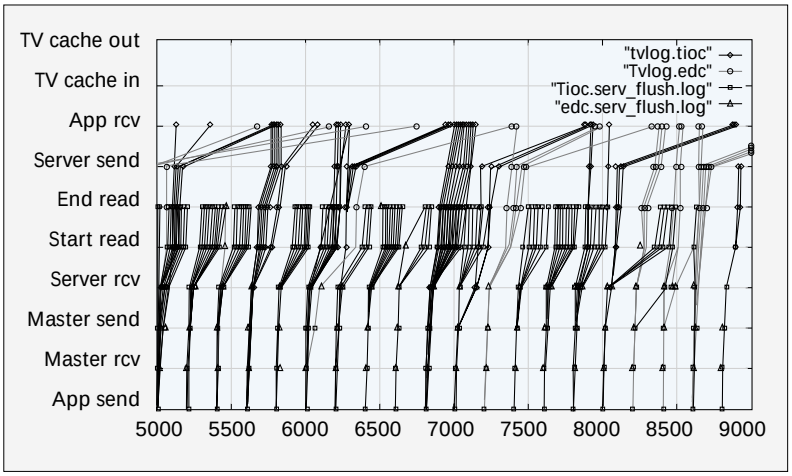


Figure 6. *Timing data from a configuration with two ISS servers. (Source: Laurence Berkeley National Laboratory)*

TerraVision is used in one of two modes, flyover or teleport, and the characteristics of the data flow for the two modes are quite different. Flyover requires a relatively steady flow over a relatively long period of time (many seconds), whereas teleport requires a large burst of data but occurs relatively infrequently. Quantitative requirements can be estimated as follows. A high-resolution full-screen display comprises about 100 tiles, each tile containing 128 x 128 pixels with 24 bits of color information, or approximately 0.4 Mb. If 10 new tiles are needed for a typical frame update during flyover, then at 30 frames/s, the average data rate is

$$(30 \text{ frames/s}) \times (10 \text{ tiles/frame}) \times (0.4 \text{ Mb/tile}) \approx 120 \text{ Mb/s}$$

at the application level in the host. Protocol overhead might add approximately 15 percent to this value, resulting in a line rate of about 140 Mb/s. For a teleport, the burst rate is considerably higher because the entire screen must be repainted within, say, a quarter of a second after the user selects the new location. If the total latency between the instant the user enters the selection and the instant the first bit of the first tile arrives at the TerraVision host is 150 ms, then the full screen of data must be transferred in the remaining (250 – 150) = 100 ms, and the capacity needed to support the transfer is

$$(100 \text{ tiles}) \times (0.4 \text{ Mb/tile})/(0.1\text{s}) \approx 400 \text{ Mb/s}$$

at the application level, or about 450 Mb/s on the transmission line.

The line capacity needed near the TerraVision host site can be determined from these required rates and from the end-to-end throughput that can be attained in the network. Measurements on the MAGIC network showed that if the MTU and the TCP window sizes were large enough, and if flow control were used, then end-to-end TCP rates corresponding to about 80 percent of the line rate could be sustained; this is about 120 Mb/s on an OC-3 line. Thus, a flyover would completely fill a single OC-3 line, so in practice two lines are needed to allow for possible degradations and for variations around the average rate derived above. Similarly, one OC-12 or four OC-3 lines are needed to support a low-response-time teleport. Lower line capacity on the path near the host would degrade the response time (although the degradation would be less than linear because of the additive factor of ISS latency). In summary, the equivalent of two OC-3 lines into the host should give satisfactory flyover performance and a teleport response time less than 0.5 s, but more capacity is needed to reduce the response time and to provide some cushion for contention near the host site.

The next question is, "How many ISS servers are needed to support the application?" Early measurements of a variety of workstations configured as ISS servers showed that a typical SCSI disk delivered data at a steady rate of about 20 Mb/s; a single SCSI adapter with multiple disks could provide about 60 Mb/s; and a workstation with multiple adapters could deliver about 80 Mb/s. Additional disks or adapters did not increase throughput — the bottleneck apparently being memory bandwidth — but did increase the probability that the throughput could be sustained by ensuring that the server was not idle. These data indicate that tiles must be distributed over at least five servers to obtain the 400 Mb/s rate needed for good teleport performance.

The data streams from the ISS servers converge at the TerraVision host, and recent measurements showed that with four servers transmitting to a host with two OC-3 ATM ports, the aggregate application-level throughput was only about 100 Mb/s, and in fact was slightly less than the throughput with a single server. (The peak throughput was about 150 Mb/s, with two input streams.) Cells are apparently being dropped at the ATM interface. This is a serious bottleneck in overall system performance; the host and interface vendors are aware of the problem and are working on a solution.

Clearly, understanding the overall performance of a network-based distributed system such as MAGIC is an appreciably more complex undertaking than simply "concatenating" the standalone performance of the individual components because there are interactions among the components. It is important to be able to measure and correlate these interactions in order to understand and predict the performance of the system as a whole. Stated in concrete terms, a problem observed by a user could have a variety of causes. For example, in MAGIC it would be acceptable if low-resolution tiles are used occasionally in place of high-resolution tiles that are delayed or lost in transit (as described in the previous section), but it would be unacceptable if this occurred frequently. If such observable degradation did occur, the cause could be the application host dropping cells, ATM switches dropping cells, excessive delay somewhere in the ISS, low ISS throughput because of the way tiles are distributed among servers, processing limitations of the TerraVision host, or a combination of these and other phenomena.

To aid in pinpointing potential problems, accurately synchronized clocks were deployed at MAGIC sites, many components were instrumented to log traffic data, and a

tool was developed for collecting, processing, and displaying the logged data [3]. The tool's graphical portrayal of measured data gives a readily comprehendible view of the overall operation of the system, permits performance estimates to be calculated easily, and provides an indication of which components may be causing performance problems. This tool, which was developed toward the end of the MAGIC project, has proved to be extremely valuable in diagnosing problems and in providing insight into techniques for improving performance. The tool is applicable to many high-speed distributed systems. A brief description of its use is given below.

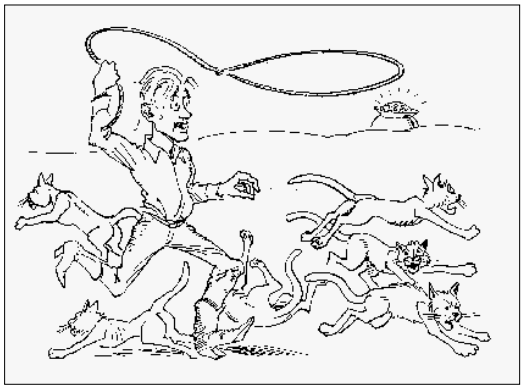


Figure 7. *In the beginning, things looked difficult.*

Figure 6 displays a representative sample of 4 s of data from a configuration with the application host in Kansas City ("tioc" in the legend), the ISS master in Sioux Falls ("edc"), and one server at each. (The host was not running TerraVision, but an application that emulates TerraVision by sending the identical tile request lists which were sent during a previously recorded TerraVision session.) The diagram traces a time history for each requested tile, showing:

- When the application sends a request list (e.g., at 6800 ms) and when it is received by the ISS master (~6810)[2]
- When the master sends tile lists to the two servers (6820) and when they are received (~6840 and ~6850)
- When the servers start and complete their read operations
- When the tile data are transmitted by the servers and received by the host

In this example, the time between the request list leaving the application host and the first tile arriving at the host is 180 ms. The diagram shows that excluding the server time, the largest component of this 180 ms delay was queuing at the server, a result of TCP retransmissions of previously transmitted tiles that were, in effect, blocking transmission of subsequent tiles. (The shallow-sloped lines between "server send" and "app receive" represent tiles with TCP transmissions.) Rough calculations of throughput at each measurement point can readily be made by counting the number of tiles processed in a selected interval of time; for example, 15 tiles were received by the application between 6980 and 7130 ms, for a throughput of about 40 Mb/s.

Herding Cats

Although the MAGIC project was an ambitious undertaking, it nevertheless was able to achieve most of its goals. The success of the project seems all the more remarkable if one considers the degree of interorganizational collaboration that was required to design, develop, test, and integrate the individual testbed components and to ensure their interoperability. Indeed, fostering this collaboration was one of the most significant nontechnical challenges facing the project — and one of its noteworthy accomplishments.

More than a dozen diverse, geographically dispersed organizations participated in MAGIC, and many of the individuals involved in the project were experienced researchers who were used to working independently. Although five of these organizations were funded by ARPA, each had its own contract and a statement of work that was complementary to the others but theoretically could be executed separately from the rest. In addition, the commercial carriers and other organizations that were expected to be major contributors were not externally funded and therefore were under no obligation to participate actively in the effort. Thus, the situation at the outset was not unlike the metaphorical herding of cats (Fig. 7).

The authors of this article were funded by ARPA to oversee and coordinate the research and development (R & D)efforts of the five ARPA-funded research participants, and to help facilitate their collaboration with the carriers and with the other organizations contributing to the project. This was a challenging assignment because none of these organizations was contractually bound to answer to a third party, so voluntary compliance of all organizations was required. Considering the cast of players and the circumstances of their affiliation, it would have been imprudent to attempt to dictate direction or to impose preferences. Furthermore, to do so not only would have been ineffective but would have been counterproductive because a heavy-handed management style would have stifled the innovation that was critical to the success of the project. In other words, peremptory management might have led to passive obedience (Fig. 8), but the results would have been uninspired [11].

The challenge was to create an environment that facilitated progress and encouraged cooperation while at the same time promoting creativity and initiative. The approach used was to obtain mutual agreement on a common set of goals and related milestones which could not be achieved without the contributions of all of the participants. In this way, the focus of the work shifted from the pursuit of individual goals to the pursuit of common goals, and collaboration was implicitly understood to be essential for success. In retrospect, the reasons why this approach worked well seem obvious. Having a common set of goals engendered an esprit de corps among the participants which gave the sense of a "virtual" organization dedicated to the success of MAGIC.

However, participants soon recognized that while camaraderie and commitment were vital to success, team spirit alone was not sufficient to ensure that success. Differences in work styles, conflicting priorities, geographical dispersion of people and resources, and the sheer magnitude of the interdependencies underscored the need for centralized leadership and for "formal" procedures for coordinating activities. As a result, members of the MAGIC team willingly consented to, and complied with, a set of management practices that they perceived as facilitating the achievement of their technical objectives. The management style was collegial with the authors serving as facilitators for defining and prioritizing project activities, as mediators for resolving disputes, as liaison with the project sponsor (ARPA), and as catalysts for promoting the team interactions required to move forward. Thus, as indicated in Fig. 9, MAGIC took a hybrid approach to managing and coordinating its R & D, with

[2] *These numerical values were obtained from a version of this diagram with an expanded timescale.*

progress toward common goals achieved through high-level consensus-building among the participants.

Three practices stand out as being most critical to the success of the project: demonstrations, planning with flexibility, and ongoing communications.

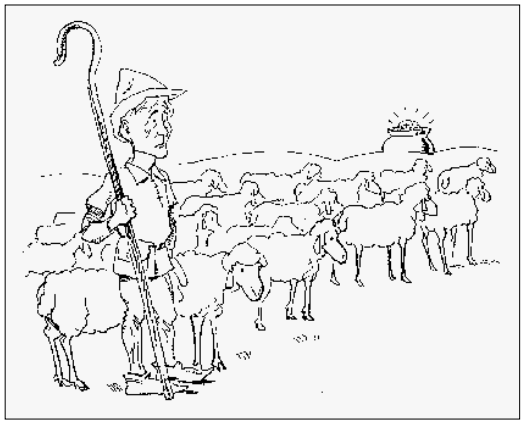


Figure 8. *This wouldn't work either.*

Demonstrations — Although the components of the MAGIC testbed were designed to operate as parts of a system, they were developed independently by organizations that were not collocated. Therefore, interoperability testing and debugging were difficult. To deal with this problem, demonstrations for external observers were scheduled to mark the achievement of milestones. These demonstrations provided a strong incentive for overcoming the logistical obstacles to testing, and for uncovering and finding solutions to tough problems. At first glance, these events appeared to be distractions from the research and a drain on people and resources, and initially they were deemed antithetical to an R & D project. In actuality they were the single most important factor in accelerating progress. Often, it was in the typically frantic last hours before a crucial demonstration that creative solutions to unforeseen problems were conceived.

A number of major demonstrations were scheduled in conjunction with quarterly project meetings or technical symposia. The first, which took place approximately halfway into the three-year project duration, marked the completion of the first phase of the MAGIC testbed: initial versions of TerraVision and the ISS working together over a partially completed backbone. The second, which occurred about six months later, demonstrated improved versions of both TerraVision and the ISS working together over the full internetwork. This demonstration was attended by prospective end users of the system who provided valuable feedback, including suggestions for additional capabilities which were subsequently incorporated into TerraVision, substantially improving the utility of the application.

Planning … with Flexibility — Researchers are notoriously reluctant to document their ideas and approaches in advance for fear of forfeiting their flexibility or limiting their options; however, failure to do so can spell disaster in a collaborative venture involving multiple organizations. Therefore, one of the first priorities of the MAGIC team was to develop a comprehensive research plan for the project. If truth be told, the process of planning was far more valuable than the plan itself. In creating the plan, each organization was forced to clearly define its tasks and milestones, to explore alternative approaches to accomplishing the work, and, most important, to identify interorganizational relationships and dependencies. It was understood that tasks and milestones, as well as technical approaches, would most likely change and evolve over the course of the three-year effort, and the plan was considered a working document to be revised and revisited as appropriate. However, at the conclusion of the project, it was gratifying to discover that the participants had accomplished most of the work they had intended to do within the allotted time and budget constraints.

Ongoing Communications — Ongoing communication was an important factor in maintaining cohesiveness among team members, and was essential for accomplishing the work. Regular interaction was achieved by holding weekly teleconferences and quarterly project meetings to discuss technical issues and interorganizational dependencies, to plan joint activities and events, and to identify and resolve problems. In addition, a variety of mechanisms for exchanging information were established, including multiple mailers, and a project server for storing and retrieving documents such as project plans, papers, and reports. To facilitate collaboration on documents, a common desktop publishing package, which was available for multiple platforms, was adopted by the team very early in the project.

Lessons Learned

The previous section described the challenges of managing the MAGIC project, and discussed some of the factors that promoted cooperation and collaboration among the participants in this multidisciplinary, multi-organizational effort. Below are some additional lessons that were learned — sometimes with pain — during the course of the three-year project.

Technology for R&D Projects

R & D projects such as MAGIC depend on state-of-the-art technology to achieve their goals. There are two alternatives for obtaining this technology: develop it as part of the project, or procure it from vendors or other sources. Where possible, MAGIC opted for the latter alternative, and milestones were planned based on vendors' stated intentions regarding the capabilities of and projected delivery dates for critical hardware and software. As a consequence of this decision, MAGIC researchers learned two important lessons.

Be Prepared to Deal with the Limitations of Vendor Products — Some of the vendor-supplied state-of-the-art products required by MAGIC, for example, the SONET terminals and the ATM switches, were available on schedule and performed satisfactorily. Others, however, were either not available in the time frame expected (e.g., OC-12 cards for the ATM switches) or did not function as anticipated. Specifically, MAGIC researchers had to deal with three types of limitations:

- Product (im)maturity: Early production versions of products required a significant amount of tuning and debugging that would be unacceptable in a mature product. For example, some workstation operating systems initially had hard-coded upper limits on the TCP window size, limiting the achievable throughput across a network having a large bandwidth-delay product.
- Standalone performance vs. system performance: Products did not perform per their standalone specifications when incorporated into a system. For example, the measured rate of a disk on an ISS server was typically less than half the specified rate (perhaps caused by interactions with the SCSI adapter).
- Single-component performance vs. multiple-component

performance: When multiple components were made to operate in parallel, their performance did not scale linearly. For example, the rate at which the TerraVision host could absorb data increased only slightly as the number of ATM interfaces was increased.

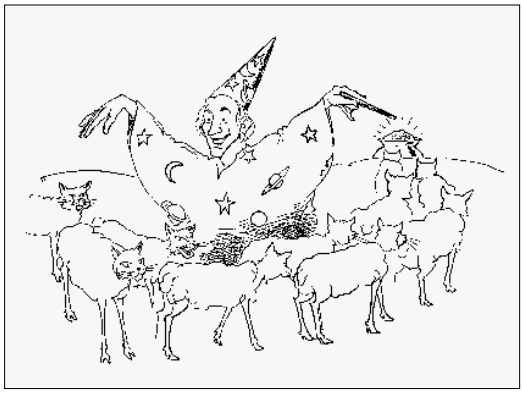


Figure 9. *Heterogeneous collaborative interoperability.*

Encourage the Active Involvement of Vendors in the R & D Effort — MAGIC depended on products that were under development or "on the horizon" when the project was initiated, and progress often hinged on timely access to early releases or upgrades. In some cases, market pressures on vendors took precedence over research needs, and the products were delayed, or anticipated features were postponed or eliminated. In other cases, products were released but were not robust, and vendor support was difficult to obtain. If equipment vendors had been more actively involved in the R & D effort, the other researchers, as well as ARPA and the carriers, would have been in a better position to influence vendor priorities and development schedules, and would have been more likely to gain the support and assistance they needed to correct shortcomings. Active vendor participation would have been beneficial to the vendors as well, providing them with insight into the strengths and limitations of their products, and helping them identify additional features and performance enhancements that might improve their competitive advantage.

Despite the difficulties associated with relying on vendors for supporting technology, using vendor-supplied products was preferable to developing customized products as part of the project. Such development would have been time- and resource-intensive, and possibly a duplication of effort. In addition, customized technology is expensive to replicate and difficult to transfer to other domains.

Support for Demonstrations

As discussed previously, demonstrations were sometimes scheduled to coincide with major project events or milestones. In addition, requests to demonstrate the capabilities of the testbed were occasionally made by ARPA, by the management of the participating organizations, or by prospective end users. While there were significant benefits associated with holding these demonstrations, preparing for them was time-consuming because it was frequently necessary to reconfigure the network and to relocate and assemble the required hardware. The MAGIC team learned two lessons that helped facilitate the conduct of demonstrations during the later stages of the project.

Establish a Reliable Testbed Configuration to Support Demonstrations — Although demonstrations proved to be a significant stimulus to progress, they sometimes conflicted with planned experiments or with development and testing activities. This was particularly troubling when work-in-progress was interrupted or put on hold for a relatively long period of time in order to reconfigure the network (or to test modifications to TerraVision or the ISS) to support a scheduled event. This situation was remedied by implementing stable versions of TerraVision and the ISS and deploying them at selected locations. These versions were used to support demonstrations, performance measurements, and related activities. Updates to the demonstration versions of TerraVision and the ISS were coordinated to ensure their compatibility.

Plan Equipment Logistics Carefully and in Advance — Another problem, pertinent to development as well as demonstrations, is the availability of equipment. Since budgets are finite, choices must be made regarding what equipment to purchase and where this equipment should be located. While it is impossible to foresee all contingencies, equipment needs should be determined as far in advance as possible. Doing so will minimize the disruptions and stress associated with disassembling and transporting hardware over long distances, and acquiring essential components on short notice. It is especially important to develop strategies for supporting off-site demonstrations, particularly those that involve relocating large, cumbersome equipment or require expensive hardware which cannot easily be moved and is difficult to borrow or lease.

One way of helping to ensure that demonstrations can be accommodated without undue disruption is to purchase spares of inexpensive equipment. These spares would be available not only for demonstrations, but for development and experimentation in the event that an original malfunctions. It is less feasible to duplicate expensive equipment; however, if vendors of critical components are actively involved in the project, they might be willing to support demonstrations by providing the necessary hardware.

Support for Development

The MAGIC testbed consists of components that were designed to interoperate but were developed independently by organizations that were geographically separated. In addition, the end users of the system were not research participants in the project. The following lessons were learned regarding how to work more effectively and efficiently under such conditions.

Build Tools to Enable Independent Development of Interoperable Components — Interoperability testing of a given component was challenging because it required that other components possess a level of functionality or performance that was not always available when the tests were ready to be conducted. One way to alleviate this problem was to implement component simulators that enabled interoperability testing. In MAGIC, the implementation of a TerraVision simulator hastened progress on the ISS, whereas the decision not to implement an ISS simulator increased the time needed to complete TerraVision.

Provide High-Speed Network Connections for All Major Participants — Proper testing of TerraVision and the ISS required high-speed interconnectivity. However, SRI and LBNL, the respective developers of these components, did not have such connectivity. As a result, interoperability testing could not be performed locally, and testing at remote sites was both burdensome and inefficient. In the MAGIC project, both of these organizations would have benefited from having high-speed links to the backbone.

Solicit Periodic Input from End Users — Getting input from end users helps to ensure that the final product has useful

features, satisfactory performance, and a well-designed interface. Input regarding desired capabilities should be solicited early in the effort and regularly thereafter as development progresses. Although, as noted in the previous section, MAGIC did benefit from such input, the project would have benefited even more if that input had been obtained earlier and more frequently.

Future Work

The MAGIC project has demonstrated a high-speed, wide-area IP/ATM internetwork that supports a real-time terrain visualization application and a distributed storage system. ARPA recently approved funding for a three-year follow-on effort, MAGIC-II, which will build on the technology developed in the original MAGIC project and on the existing MAGIC network facilities. There are two major interrelated goals in MAGIC-II:

- To enhance and upgrade the testbed to demonstrate the utility and capabilities of distributed processing and network-based storage, coupled with high-speed networks, to support a new generation of real-time applications.
- To create a very large internetwork with many end users that will be a realistic test environment for ATM technology and for the above type of application.

The MAGIC-II testbed will demonstrate the scalability of the distributed storage and distributed processing concepts by configuring systems that have a large number of servers and processors on many ATM networks spanning a large geographic area, and have multiple sets of data and multiple simultaneous users.

The MAGIC-II testbed will demonstrate the scalability of the distributed storage and distributed processing concepts by configuring systems that have a large number of servers and processors on many ATM networks spanning a large geographic area, and have multiple sets of data and multiple simultaneous users.

The MAGIC-II testbed is based on a very general paradigm in which high-performance computing, storage, and communications are used to provide rapid access to large amounts of distributed data, including real-time data that must be processed and delivered to an end user on demand. Applications that use this paradigm arise in a variety of situations, including military operations, intelligence imagery analysis, and natural disaster response. The exact type, location, and ownership of the data used by these applications may not be known in advance, and these data may require a large amount of processing to be transformed into useful information. In addition, the processed data may have to be delivered to end users with a range of communications speeds, link qualities, computational powers, and display capabilities. The data, as well as the computing and storage resources required to process them, may reside in multiple administrative domains that have different usage and access control policies.

Specific work to be done in MAGIC-II includes augmenting the MAGIC internetwork with wireless nodes and interconnecting it with other IP/ATM internetworks to create a nation-wide, high-speed, wide-area testbed. This testbed will be used for experimentation with protocols, with routing techniques, and with mobile access to backbone services. A new version of TerraVision that can perform on-the-fly rectification coupled with algorithms for "in-transit" processing will permit near-real-time visualization of raw imagery, enabling data from sensors to be viewed within minutes (rather than hours) after being generated. (Fig. 1.) Data fusion techniques will allow disparate data types to be overlaid. The processing will be performed by sets of distributed devices that are constructed from resources owned by multiple administrative domains. Algorithms that dynamically determine the current state of the network will provide information to the application so that it can adapt to current system performance and to available system resources.

The MAGIC-II project will certainly benefit from the lessons learned in the original MAGIC project. Nevertheless, as with any research effort, new challenges will be encountered, and new lessons, both technical and organizational, will be learned in meeting these challenges. Stay tuned.

References

[1] R. Binder, "Issues in Gigabit Networking," *Proc. IEEE Globecom '92*, Orlando, FL, 12/92. Also see: http://www.CNRI.Reston.VA.US:4000/public/overview.html

[2] J. Evans *et al.*, "A 622 Mb/s LAN/WAN Gateway and Experiences with Wide Area ATM Networking," *IEEE Network*, this issue.

[3] B. Tierney *et al.*, "Performance Analysis in High-Speed Wide Area IP-over-ATM Networks: Top-to-Bottom End-to-End Monitoring," *IEEE Network*, this issue.

[4] M. Laubach, "Classical IP and ARP over ATM," RFC 1577; 20, Jan. 1994.

[5] B.J. Ewy *et al.*, "TCP/ATM Experiences in the MAGIC Testbed," *Proc. 4th IEEE Symp. High Perf. Dist. Comp.*, Aug.1995, pp. 87–93.

[6] J. D. Cavanaugh, Minnesota Supercomputer Center, personal communication.

[7] Y. G. Leclerc and S. Q. Lau Jr., "TerraVision: A Terrain Visualization System," Tech. Note 540, SRI International, Menlo Park, CA, Apr. 1994.

[8] B. Tierney *et al.*, "Distributed Parallel Data Storage Systems," *Proc. ACM Multimedia '94*, Oct. 1994. Also available as http://george.lbl.gov/ ISS/papers/ISS-paper.ACM.final.html

[9] L. T. Chen and D. Rotem, "Declustering Objects for Visualization," *Proc. 19th VLDB (Very Large Database) Conf.*, 1993.

[10] B. Tierney *et al.*, "Using High Speed Networks to Enable Distributed Parallel Image Server Systems," *Proc. Supercomputing '94*, Nov. 1994. Also available as: http://george.lbl.gov/ ISS/papers/ISS-paper.SC94.final.html

[11] S. Adams, "Dilbert," *Boston Globe*, Jan. 14, 1996.

Biographies

BARBARA FULLER recently joined Mitretek Systems in McLean, Virginia, as lead staff in the Center for Information Technology Systems. Prior to joining Mitretek, she held a similar position with the MITRE Corporation in Bedford, Massachusetts, where she provided technical, systems engineering, and program planning, coordination, and analysis support to a variety of DoD-sponsored advanced information systems projects, including MAGIC. She also spent 12 years at MITRE managing multidisciplinary projects for U.S. Government agencies dealing with toxic chemicals in the environment. She received her B.A. degree in chemistry from Western Reserve University in Cleveland, Ohio, and her M.S. and Ph.D. degrees in chemistry from New York University.

IRA RICHER is director of networking research at the Corporation for National Research Initiatives (CNRI), Reston, Virginia. His current work includes coordinating the activities of about a dozen companies in a project involving trials of broadband services to residences, and managing the MAGIC project. Prior to joining CNRI, Dr. Richer was with MITRE, where he supervised a small group working on advanced networks and applications. From 1988 to 1991, Dr. Richer was program manager for high-performance networking at ARPA. He initiated ARPA's program in gigabit networks, and he launched ARPA's work in all-optical networking. Dr. Richer received a B.E.E. degree from Rensselaer Polytechnic Institute, and M.S. and Ph.D. degrees from the California Institute of Technology.