6 GI System Software

LEARNING OBJECTIVES

After studying this chapter you will be able to:

- Understand the architecture of GI systems, specifically
  – Organization by project, department, and enterprise
  – The three-tier architecture of software systems (graphical user interface, tools, and data access).
- Outline how to customize GI system products.
- Describe the main types of commercial software:
  – Desktop
  – Web mapping
  – Server
  – Virtual globe
  – Developer
  – Mobile
  – Other.
- Describe the key characteristics of the main types of commercial GI system products currently available.

At the core of an operational geographic information system lies some computer software that forms the basis of the processing engine. A geographic information (GI) system comprises an integrated collection of computer programs that implement geographic storage, processing, and display functions. The three key parts of any GI system are the user interface, the tools (functions), and the data manager. All three parts may be located on a single computer, or they may be spread over multiple machines in a departmental or an enterprise system configuration. Three main types of architecture are used to build operational GI system implementations: desktop, client server, and Cloud. There are many different types of GI system, and this chapter organizes the discussion around the main categories: desktop, Web mapping, server, virtual globe, developer, mobile, and other. Software may be licensed using a commercial or open-source model. Examples of the main types of software products from the leading commercial developers include Autodesk AutoCAD, Bentley Map, Esri ArcGIS, and Intergraph GeoMedia. There is also a significant and growing open-source GI software community, of which Quantum GIS, Map Server, and the OpenGeo Suite are well-known examples.

6.1 Introduction

In Chapter 1, the technical parts of a GI system were defined as the network, hardware, software, and data, which together functioned with reference to people and the procedural frameworks within which people operate (Figure 1.12). This chapter is concerned with software: the geographic storage, processing, and display engines of a complete, working GI system. The functionality or capabilities of GI systems will be discussed later in this book (especially in Chapters 8–15). The focus here is on the different ways in which these capabilities are realized in software products and implemented in operational GI systems.

This chapter takes a fairly narrow view of GI systems, concentrating on a range of generic capabilities to collect, store, manage, query, analyze, and present geographic information. It excludes atlases, simple graphics and mapping systems, route-finding software, simple location-based services, image-processing systems, and spatial extensions to database management systems (DBMS), which are not true GI systems as defined here. The discussion is also restricted to GI system products—well-defined collections of software and accompanying documentation, install scripts, and so on—that are subject to multiversioned release control (that is to say, the software is made available in a controlled way on a periodic basis). By definition, it excludes specific-purpose utilities, unsupported routines, and ephemeral codebases.

Earlier chapters, especially Chapter 3, introduced several fundamental computer concepts, including digital representations, data, and information. Two further concepts need to be introduced here. Programs are collections of instructions that are used to manipulate digital data in a computer. System software, such as a computer operating system, is used to support application software—the programs with which end users interact. Integrated collections of application programs are referred to as software packages or systems (or just software for short).

Software can be distributed to the market in several different ways. Increasingly, download from the Internet is the main distribution mechanism, although hard-copy media (DVDs) are still widely used. GI system products usually comprise an integrated collection of software programs, and a selection of additional elements such as an install script, online help files, sample data and maps, documentation, a license, and an associated Web site. Alternative distribution models that are becoming increasingly prevalent for smaller products include *shareware* (usually intended for sale after a trial period), *liteware* (shareware with some capabilities disabled), *freeware* (free software but with copyright restrictions), *public domain software* (free with no restrictions), and *open-source* software (the *source code* is provided under an open-source license arrangement, and users agree not to limit the distribution of improvements). In the early days open-source software products provided only rather simple, poorly engineered tools with no user support. Today there are several high-quality, feature-rich software products, a number of which will be discussed in later sections.

GI system software packages provide a unified approach to working with geographic information.

GI system software is built on top of basic computer operating system capabilities that deal with such things as security, file management, peripheral drivers (controllers), Web interaction, printing, and display management. GI system software is constructed on these foundations to provide a controlled environment for GI collection, management, analysis, display, and interpretation. The unified architecture and consistent approach to representing and working with geographic information in a GI system both aim to provide users with a standardized way of working with geographic information.

6.2 The Evolution of GI System Software

In its formative years, GI system software consisted simply of collections of computer routines that a skilled programmer could use to build an operational GI system. During this period, each and every GI system was unique in terms of its capabilities, and significant levels of resource were required to create a working system. As software engineering techniques advanced and the GI system market grew in the 1970s and 1980s, demand increased for higher-level applications with a standard user interface. In the late 1970s and early 1980s, the standard means of communicating with a GI system was to type in command lines and wait for a response from the system. User interaction with a GI system entailed typing instructions, for example, to draw a topographic map, query the attributes of a forest stand object, or summarize the length of highways in a project area. Essentially, a GI system was a toolbox of geoprocessing operators or commands that could be applied to datasets to create new derivative datasets. For example, three polygon-based data layers of the same geographic area—*Soil*, *Slope*, and *Vegetation*—could be combined using an overlay processing function (see Section 13.2.4) to create an *IntegratedTerrainUnit* dataset.

Two key developments in the late 1980s and 1990s made the software easier to use and more generic. First, command line interfaces were supplemented and eventually largely replaced by graphical user interfaces (GUIs). These menu-driven, form-based interfaces greatly simplify user interaction with a GI system (see Section 12.2.1). Second, a customization capability was added to allow specific-purpose applications to be created from the generic toolboxes. Software developers and advanced technical users could make calls using a high-level programming language (such as Visual Basic, C, or Java) to published application programming interfaces (APIs) that exposed key functions. Together these stimulated enormous interest in GI systems and led to much wider adoption and expansion into new areas.

In particular, the ability to create custom application solutions allowed developers to build focused applications for end users in specific market areas. This led to the creation of GI applications specifically tailored to the needs of major markets (e.g., government, utilities, military, and environment). New terms were developed to distinguish these subtypes of GI system: planning information systems, automated mapping/facility management (AM/FM) systems, land information systems, and more recently, location-based services.

In the past few years a new method of software interaction has evolved that allows systems to communicate over the Web using a Web services paradigm. A Web service is an application that exposes its functions via a well-defined published interface that can be accessed over the Web from another program or Web service. This new software interaction paradigm allows geographically distributed GI system functions to be linked to create complete GI applications (Box 6.1). Figure 6.1 shows the program code used to draw a map of Sydney in Google Maps. For example, a market analyst who wants to determine the suitability of a particular site for locating a new store can start a small interactive program on his or her smartphone or tablet device that links to remote services over the Web that provide access to the latest population census and

Technical Box 6.1

Cloud GI Web Services: Requesting a Google Map

GI applications that operate over the Web often use data and functionality hosted in servers managed in a Cloud computing environment. The geographic data and processing functions are accessible via APIs. Comparatively lightweight applications running on devices such as phones and tablets, or within browsers, can make calls to an API to perform actions such as request a map, geocode an address, add new data points, and route between locations. JavaScript is a widely used simple programming and API environment for mapping and analysis (and other) applications. Figure 6.1 shows a script running inside a browser that requests a simple map from the Google Maps Web service that is hosted in Google's Cloud (a set of servers in a data center accessible over the Web).

JavaScript is an XML-based language that encodes instructions as tags (pairs of instructions within angled brackets <>).

Map source: Google, map data copyright Google

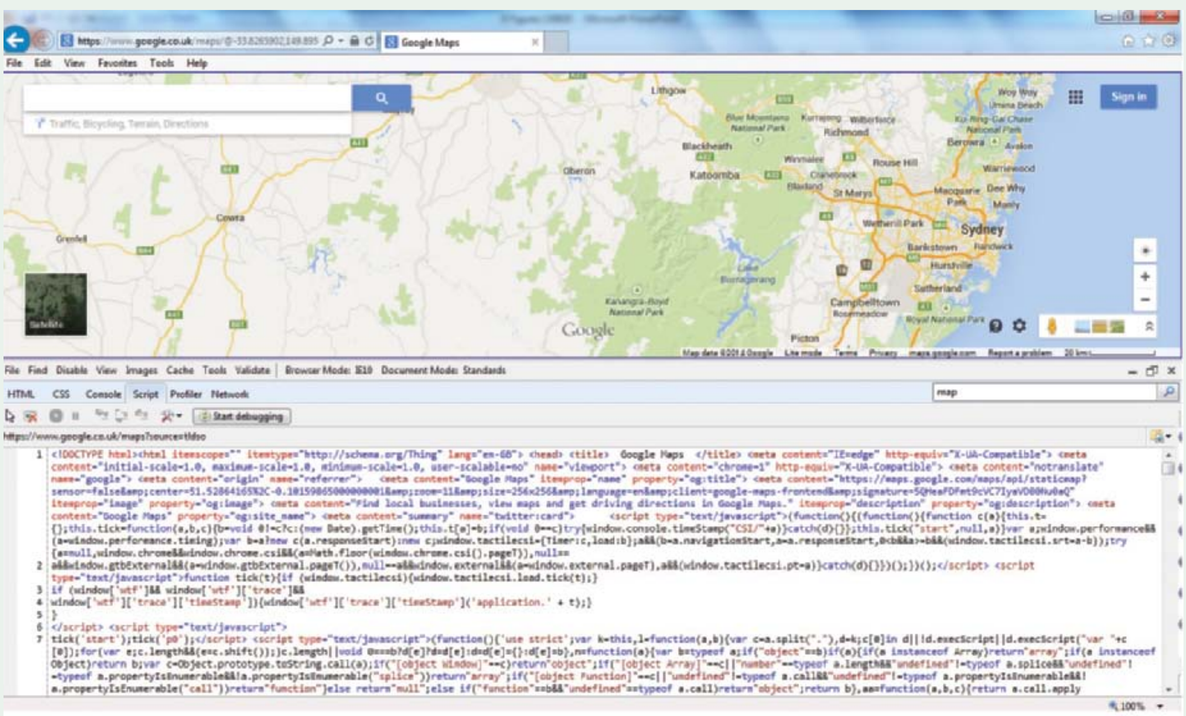


Figure 6.1 Google Maps Web service: Web browser view of map and part of script.

geodemographics data, as well as analytical models. Although these data and programs are remotely hosted and maintained, they can be used for site suitability analysis as though they were resident on the market analyst's device. Chapter 10 explores Web services in more depth in the context of distributed GI systems. This software architecture for Web services is one of the key technology building blocks of Cloud GI systems (GI services hosted in data centers that can be accessed over the Web).

Devices connected to Cloud GI systems are becoming increasingly popular as an implementation architecture for GI technology.

6.3 Architecture of GI System Software

6.3.1 Project, Departmental, and Enterprise GI Systems

Major GI systems can be implemented using three main scales of software implementation: project, departmental, and enterprise.

In project implementations the technical components (network, hardware, software, and data) of an operational GI system are assembled for the duration of a project, which may be from several days, to months or a few years. Data are collected or downloaded from the Web specifically for the project, and little thought is usually given to reuse of software, data, and human knowledge. In larger organizations, multiple projects may run one after another or even in parallel. The "one off" nature of the projects, coupled with an absence of organizational vision, often leads to duplication, as each project develops using different hardware, software, data, people, and procedures. Sharing data and experience is usually a low priority.

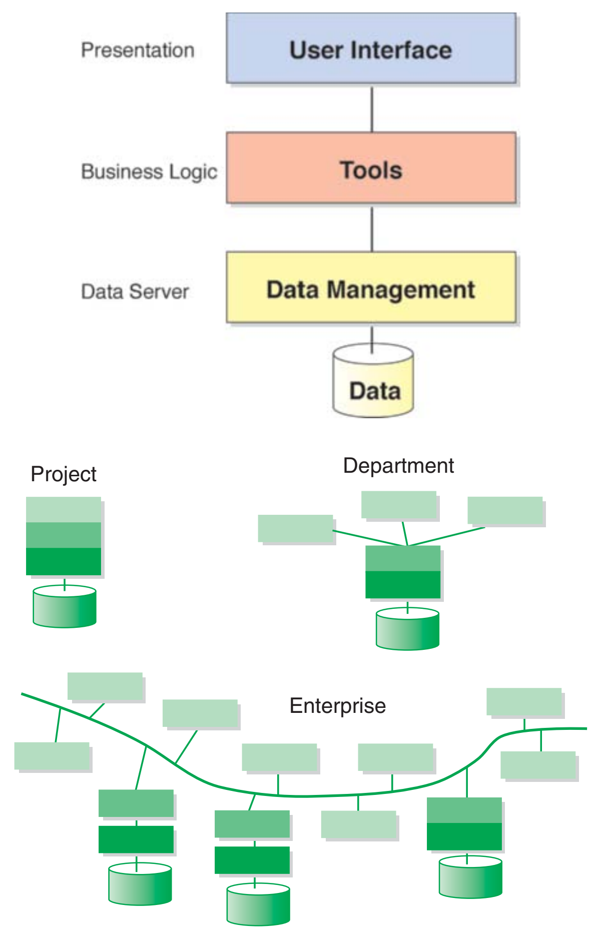
As interest in GI systems grows, to save money and encourage sharing and resource reuse, several projects in the same department may be amalgamated. This often leads to the creation of common standards, development of a focused GI team, and procurement of new GI system capabilities. Yet it is also quite common for different departments to have different GI systems and standards.

As GI systems become more pervasive, organizations learn more about them and begin to become dependent on them. This leads to the realization that GI systems are a useful way to organize many of an organization's assets, processes, and workflows. Through a process of natural growth, and possibly further major procurement (e.g., purchase of upgraded hardware, software, and data), GI systems gradually become accepted as important enterprise-wide information systems. At this point, GI standards are accepted across multiple departments, and resources to support and manage the GI systems are often centrally funded and managed. Of course, things do not always proceed as smoothly as this and in such a linear way.

6.3.2 The Three-Tier Architecture

From an information systems perspective a GI system has three key parts: the user interface, the tools, and the data management system (Figure 6.2). The user's interaction with the system is via a graphical user interface (GUI; see also Section 12.2.1), an integrated collection of menus, tool bars, and other controls. The GUI provides organized access to tools for handling GI. The tool set defines the capabilities or functions that the GI system has available for processing geographic data. The data are stored as files, databases, or Web services and are organized by data management software. In standard information-system terminology this is a three-tier architecture, with the three tiers being

Figure 6.2 Classical three-tier architecture of a GI system.



called presentation, business logic, and data server. Each of these software tiers is required to perform different types of independent tasks. The presentation tier must be adept at collecting user inputs, rendering (displaying) data, and interacting with graphic objects. The business logic tier is responsible for performing all operations such as network routing, data overlay processing, and raster analysis. It is here also that the GI data-model logic is implemented (see Section 6.3.3 for discussion of data models). The data-server tier must import and export data and service requests for subsets of data (queries) from a database or file system. In order to maximize system performance, it is useful to optimize hardware and operating system settings differently for each type of task. For example, rendering maps requires large amounts of memory and fast CPU speeds, whereas database queries need fast disks and buses (interfaces between devices) for moving large numbers of data around. By placing each tier on a separate computer, some tasks can be performed in parallel, and greater overall system performance, scalability, and resilience can be achieved.

GI systems deal with user interfaces, tools, and data management.

Three main types of computer system architecture are used to build operational GI systems: desktop, client–server, and Cloud. In the simplest desktop configuration, as used in a single-user project, the three software tiers are all installed on a single piece of hardware (most commonly a desktop PC) in the form of a desktop GI system, and users are usually unaware of their separate identity (Figure 6.3A). In a variation on this theme, data files are held on a PC (or sometimes a Windows, Linux, or Unix machine) centralized file server, but the data–server functionality is still part of the desktop GI system. This means that the entire contents of any accessed file must be pulled across the network, even if only a small amount of it is required (Figure 6.3B).

In larger and more advanced multiuser workgroup or departmental GI systems, the three tiers can be installed on multiple machines to improve flexibility and performance (Figure 6.4). In this type of configuration, the users in a single department (for example, the planning or public works department in a typical local government) still interact with a GUI (presentation layer) on their desktop computers, which also contain all the business logic, but the database management software (data server layer) and data may be located on another machine connected over a network. This type of computing architecture is usually referred to as client–server because clients request data or processing services from servers that perform work to satisfy client requests. The data server has local processing capabilities and is able to query and process data and thus return part of the whole database to the client, which is much more efficient than sending back the whole dataset for client-side query. Clients and servers can communicate over a local area network (LAN), wide area network (WAN), or Internet network, but given the large amount of data communication between the client and server, faster LANs are most widely used.

Both the desktop and client–server architecture configurations have significant amounts of functionality on the desktop, which is said to be a thick client (see also Section 6.3.4).

In a client–server GI system, clients request data or processing services from servers that perform work to satisfy client requests.

Figure 6.3 Desktop GI system architecture: (A) standalone desktop system on PCs each with own local files; (B) Desktop system on PCs sharing files on a PC file server over a LAN.

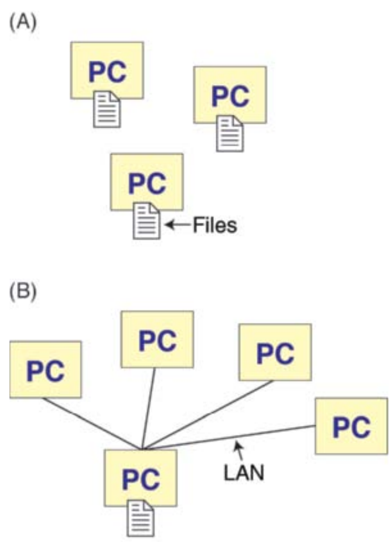
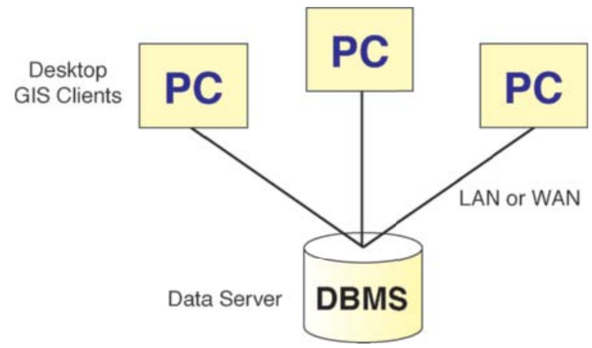


Figure 6.4 Client–server system: desktop software and DBMS data server in a workgroup or departmental configuration.



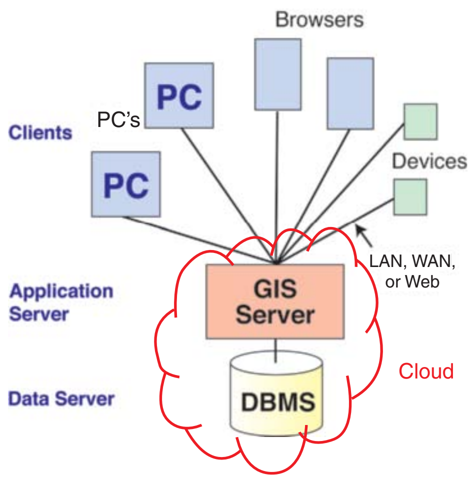


Figure 6.5 Centralized server system as used in advanced departmental and enterprise implementations. See Figure 6.2 for explanation of color-coding.

In recent years Cloud GI systems have become popular largely because of the ease of use and cost-effective nature of this architecture. The key feature of a Cloud implementation is that almost everything runs on servers that are hosted remotely in data centers that are accessible over the Web (Figure 6.5). In this architectural pattern all computing and geographic data and processing resources run on third-party hardware and software. Figure 6.6 shows the computer hardware in a typical data center. Such computing resources are usually purchased accordingly to usage (e.g., CPU and disk space utilized per month) and can be quickly provisioned (scaled up or down) depending on usage requirements.

In the Cloud model, users access business logic and data services from devices (e.g., phones or tablets), which run lightweight clients (thin clients), or from a range of heavier-weight clients, including full-functionality desktop GI systems (thick clients). In the case of thin-client access, the presentation tier (user interface) also runs in the Cloud on the server (although technically it is still presented on the desktop). The current trend is to add local processing capabilities to thin clients by supporting client-side scripting and data caching. A second server machine is usually employed to run data management software (DBMS). This type of implementation is the standard architecture for Web-based GI systems and is common in enterprise GI systems. Large, enterprise GI systems may involve more than 10 servers and hundreds or even thousands of clients that are widely dispersed geographically and nowadays are connected using the Web. The Cloud computing architecture forms the platform for many Web-based services such as the e-mail and search systems from Google, Microsoft, and Yahoo!, as well as various virtual-globe implementations (see Section 6.6.4).

Figure 6.6 Computer hardware in a typical data center.

Cloud computing is the standard architecture used in Web-based and enterprise GI systems.

Although organizations often standardize on either a project, a departmental, or an enterprise system, it is also common for large organizations to have all three configurations operating in parallel or as subparts of a full-scale system.

6.3.3 Software Data Models and Customization

In addition to the three-tier model, two further topics are relevant to an understanding of software architecture: data models and customization.

GI data models will be discussed in detail in Chapter 7, and so the discussion here will be brief. From a software perspective, a data model defines how the real world is represented in a GI database. It will also affect the type of software tools (functions or operators) that are available, how they are organized, and their mode of operation. A data model defines how the different tools are grouped together, how they can be used, and how they interact with data. Although such issues are largely transparent to end users whose interaction with a GI system is via a user interface, they become very important to software developers that are interested in customizing or extending software. The data model affects the capability, flexibility, and extensibility of GI system software.

Customization is the process of modifying a GI system to, for example, add new functionality to applications, embed GI functions in other applications, or create specific-purpose applications. It can be as simple as deleting unwanted controls (for example, menu choices or buttons) from a GUI, or as sophisticated as adding a major new extension to a software package for such things as network analysis, high-quality cartographic production, or land parcel management.

To facilitate customization, GI software must provide access to the data model and expose capabilities to use, modify, and supplement existing functions. In the late 1980s when customization options were first added to GI software products, each vendor had to provide a proprietary customization capability simply because no standard customization systems existed. Nowadays, with the widespread adoption of the .Net and Java frameworks, a number of industry-standard approaches and programming languages (such as Visual Basic, C/C^{++}/C#, Java, and Python) are available for customizing GI systems. Simpler scripting languages such as JavaScript are also becoming widely used, especially for Web services, because of their high-level, interpreted nature and because a number of different libraries (e.g., for charting, database query, and search) can easily be integrated into the same end-user applications.

Modern programming languages are one component of larger developer-oriented packages called integrated development environments (IDEs). This term refers to the combination of several software development tools, including a visual programming language, an editor, a debugger, and a profiler. Many of the so-called visual programming languages, such as C^{++}/C#, Visual Basic, and Java, support the development of windows-based GUIs containing forms, dialogs, buttons, and other controls. Program code can be entered and attached to the GUI elements using the integrated code editor. An interactive debugger will help identify syntactic problems in the code, for example, misspelled commands and missing instructions. Finally, there are also tools to support profiling programs. These show where resources are being consumed and how programs can be speeded up or improved in other ways.

Contemporary GI systems typically use an industry-standard programming language like Visual Basic, C^{++}/C#, or Java for customization.

To support customization using open, industry-standard IDEs, GI system functions must be exposed in a standard way. In modern GI systems the functionality is developed as software components—self-contained software modules that implement a coherent collection of functionality. A key feature of such software components is that they have well-defined application programming interfaces (APIs) that allow the functionality to be called by the programming tools in an IDE or scripting environment.

Three predominant technology standards are widely used for defining and reusing software components. For building high-performance, interactive desktop applications, Microsoft's .Net framework is the de facto standard. It uses fine-grained components (that is, a large number of small functionality blocks). For server-centric GI systems, both .Net and the equivalent Java frameworks are widely deployed in operational applications. Although both .Net and Java work very well for building fine-grained client or server applications, they are less well suited for building applications that need to communicate over the Web. Because of the loosely coupled, comparatively slow, heterogeneous nature of Web networks and applications, fine-grained programming models have limitations. As a consequence, coarse-grained component and messaging systems have been built on top of the fine-grained .Net and Java applications using SOAP/XML (simple object access protocol/extensible markup language) and JavaScript/REST (representation state transfer protocol) that allow applications with Web services interfaces to interact over the Web. Microsoft's Silverlight and Adobe's Flex are also being widely used to build Web applications. This higher-level approach makes it easier to create custom Web applications that use Web resources in a more efficient way by minimizing network traffic.

Components are important to software developers because they are the mechanism by which reusable, self-contained, software building blocks are created and used. They allow many programmers to work together to develop a large software system incrementally. The standard open (published) format of components means that they can easily be assembled into larger systems. In addition, because the functionality within components is exposed through interfaces, developers can reuse them in many different ways, even supplementing or replacing functions if they so wish. Users also benefit from this approach because GI system software can evolve incrementally and support multiple third-party extensions. In the case of GI systems this includes, for example, tools for charting, reporting, and data table management.

Box 6.2 describes the work of one open-source software developer, Gabriel Roldan (Figure 6.7).

6.3.4 GI Systems on the Desktop, on the Web, and in the Cloud

Today mainstream, high-end GI system users work primarily with software that runs either on the desktop or over the Web and in the Cloud. In the desktop case, a PC (personal computer) is the main hardware platform, and Microsoft Windows remains the dominant

Biographical Box 6.2

Gabriel Roldan, Open-Source Software Developer

Gabriel has been an open-source GI system developer for over a decade and is a core OpenGeo developer. OpenGeo is the commercial developer of the OpenGeo Suite of open-source geospatial software. This is a geospatial Web services stack for deploying solutions for Web mapping, transportation, telecommunications, open government, and a huge range of other solutions. The OpenGeo Suite is a continually updated geo Web services platform along with maintenance agreements, which include support and training. OpenGeo is funded by leading investment firms Vanedge Capital and IQT.

Gabriel began in the GIS and Remote Sensing Department in the Instituto Politecnico Superior in Rosario, Argentina, and since then he has worked at various companies in Argentina and Spain on route finding, vehicle tracking, and Web mapping, improving corporate GI and spatial data infrastructure software. Working with local governments, he has introduced GeoTools and GeoServer and helped with their adoption as core components of GI software infrastructures.

Courtesy: Gabriel Roldan

Figure 6.7 Gabriel Roldan, open-source software developer.

Gabriel joined the open-source community in 2003 and has developed the ArcSDE datastore and Web Mapping Service for GeoServer as well as the first ArcSDE data reader, which he still maintains. At OpenGeo, he rearchitected GeoTools' ArcSDE raster support to match ArcSDE's capabilities and became a core GeoServer and GeoWebCache developer. Since 2011 he has been the technical lead for the Geogit project, which is implementing database versioning in GeoServer.

"Working in professional open-source is an incredible opportunity for continuous learning and improvement, both technically and as a human being," says Gabriel.

operating system (Figure 6.8 and Table 6.1). In the desktop paradigm, clients tend to be functionally rich and substantial in size and are often referred to as thick or fat clients. Using the Windows standard facilitates interoperability (interaction) with other desktop applications, such as word processors, spreadsheets, and databases. As noted earlier (Section 6.3.2), most sophisticated and mature GI workgroups have adopted the client–server implementation approach by adding either a thin or thick server application running on the Windows, Linux, or Unix operating system. The terms *thin* and *thick* are less widely used in the context of

Figure 6.8 Desktop and network GI system paradigms. See Figure 6.2 for explanation of color coding.

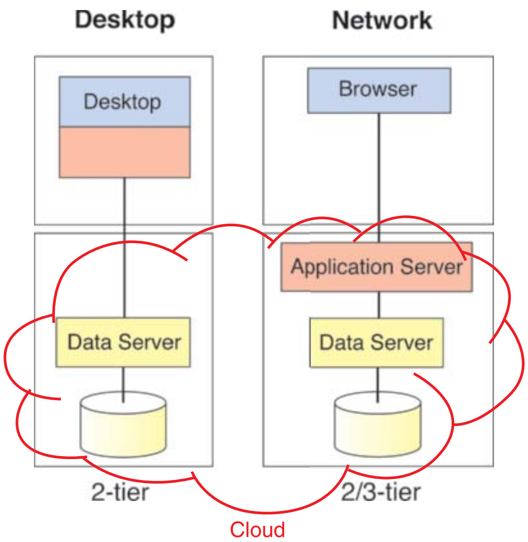


Table 6.1 Comparison of desktop and network GI systems.

Feature	Desktop	Network
Client size	Thick	Thin
Client platform	Windows	Browser or Device
Server size	Thin/thick	Thick
Server platform	Windows/Unix/Linux	Windows/Unix/Linux
Component standard	.Net	.Net/Java
Network	LAN/WAN	LAN/WAN/Internet

servers, but they mean essentially the same as when applied to clients. Thin servers perform relatively simple tasks, such as serving data from files or databases, whereas thick servers also offer more extensive analytical capabilities such as geocoding, routing, mapping, and spatial analysis. In desktop implementations, LANs and WANs tend to be used for client–server communication. It is natural for developers to select Microsoft's .Net technology framework to build the underlying components making up these systems given the preponderance of the Windows operating system, although other component standards could also be used. The Windows-based client–server architecture is a good platform for hosting interactive, high-performance GI applications. Examples of applications well suited to this platform include those involving geographic data editing, map production, 2-D and 3-D visualization, spatial analysis, and modeling. It is currently the most practical platform for general-purpose systems because of its wide availability, good performance for a given price, and common usage in business, education, and government.

GI system users are standardizing their systems on the desktop and Web implementation models.

In the past few years there has been increasing interest in harnessing the power of the Web for GI systems and for hosting applications in a cloud computing environment. Although desktop GI systems have been and continue to be very successful, users are constantly looking for lower costs of ownership and improved access to geographic information. Network-based (sometimes called distributed) GI systems allow previously inaccessible information resources to be made more widely available. A particularly interesting type of network GI system is Web-based, using the Web as a network and Cloud infrastructure for managing resources. The network model intrigues many organizations because it is based on centralized software and data management, which can dramatically reduce initial implementation and ongoing support and maintenance costs. It also provides the opportunity to link nodes of distributed user and processing resources using the medium of the Internet. The continued rise in network GI systems will not signal the end of desktop GI systems—indeed quite the reverse is true because it is likely to stimulate the demand for content and professional skills in geographic database automation and administration and applications that are well suited to running on the desktop.

In contrast to desktop systems, networked GI systems can use cross-platform Web browsers or devices like smartphones and tablets to host the viewer user interface. Currently, clients are typically very thin, often with simple display and query capabilities, although there is an increasing trend for them to become more functionally rich. Server-side functionality may be encapsulated on a single server, although in medium and large systems it is more common to have two servers, one containing the business logic (a middleware application server), the other the data manager (data server). The server applications typically contain all the business logic, are comparatively thick, and may run on a Windows, Unix, or Linux platform.

Recently, there has been a move to combine the best elements of the desktop and network paradigms to create so-called rich clients. These are stored and managed on the server and are dynamically downloaded to a client computer according to user demand. The business logic and presentation layers run on the server, with the client hardware simply used to render the data and manage user interaction. The new software capabilities in recent editions of the .Net and Java software development kits allow the development of applications with extensive user interaction that closely emulate the user experience of working with desktop software. Microsoft's Bing Maps 3D is an example of a rich client. When a user visits the Bing Web site, he or she is given the opportunity to download the Bing Maps 3D client. Upon installation, the client talks to the Bing Maps 3D server that holds vast databases of geographic information for the Earth and provides server-side query and analysis features.

6.4 Building GI Software Systems

Widely used GI systems are built and released as software products by development and product teams that may operate according to commercial or open-source models. Mature software products are subject to carefully planned versioned release cycles that incrementally enhance and extend the pool of capabilities. The key parts of a GI system architecture—the user interface, business logic (tools), data manager, data model, and customization environment—were outlined in the previous section.

GI product teams start with a formal design for a software system and then build each part or component separately before assembling the whole system. Typically, development will be iterative with creation of an initial prototype framework containing a small number of partially functioning parts, followed by increasing refinement of the system. Core software systems are usually written in a modern programming language like Visual C^{++}/C# or Java, with Visual Basic or another scripting language such as Python used for operations that do not involve significant amounts of computer processing like GUI interaction.

As standards for software development become more widely adopted, so the prospect of reusing software components becomes a reality. A key choice that then faces all software developers or customizers is whether to design a system by reusing third-party components or libraries (that might be licensed commercially or using an open-source distribution model), or to build one more or less from scratch. All three options have advantages and disadvantages: building components gives greater control over capabilities and enables specific-purpose optimization, but can be complex and slow; buying components can save time, but can be expensive and inflexible; the open-source model provides low-cost access to functionality and the ability to view and control source code, but often lacks ongoing support.

A key GI system implementation issue is whether to buy a system or to build one.

A modern GI system comprises an integrated suite of software components of three basic types: data management software for controlling access to data (Chapters 7, 8, and 9); mapping software for display and interaction with maps and other geographic visualizations (Chapters 11 and 12); and spatial analysis and modeling software for transforming geographic data using operators (Chapters 13, 14, and 15). The components for these parts may reside on the same computer or can be distributed widely (Chapter 10) over a network.

6.5 GI Software Vendors

The GI industry is fortunate to have a diverse range of significant and well-established software vendors. Each of the major vendors brings its own heritage and experience and has created unique elements and interpretations of core capabilities in the form of software that is constantly evolving in response to ongoing IT changes and expanded user requirements. The following sections briefly describe the main commercial GI software vendors. Space does not permit similar treatment of smaller open-source development teams, although the work of one open-source developer is discussed in Box 6.2.

6.5.1 Autodesk

Autodesk is a large and well-known publicly traded company with headquarters in San Rafael, California. It is one of the world's leading digital design and content companies and serves customers in markets where design is critical to success: building, manufacturing, infrastructure, digital media, and location services. Autodesk is best known for its AutoCAD product family, which is used worldwide by more than 4 million customers. The company was founded more than 30 years ago and has grown to become a $2 billion entity employing over 7,500 staff.

Autodesk's product family for the GI marketplace features several products including AutoCAD Map 3D, Autodesk Infrastructure Map Server, and Autodesk Infrastructure Design Server. AutoCAD Map 3D is a desktop product that enables organizations to create, maintain, and visualize geographic data and is especially adept at bridging the gap between engineering and GI teams on the one hand, and the rest of an organization on the other. It is very widely used for data capture in architecture, engineering, and construction firms, as well as local-government engineering departments. Autodesk Infrastructure Map Server is used to publish and share CAD, GI, and asset information with Web-based mapping software. It has capabilities to access, visualize, and coordinate enterprise data with workers in the field, operations and customer service personnel, and the public by using Web-based mapping dashboards and reports. Autodesk Infrastructure Design Suite is a design solution that combines intelligent, model-based tools to gain accurate, accessible, and actionable insight in transportation, land, and water projects.

6.5.2 Bentley

Bentley, a privately held developer of software solutions for the infrastructure lifecycle market, is headquartered in Exton, Pennsylvania. It has applications that help engineers, architects, contractors, governments, institutions, utilities, and owner–operators design, build, and operate productively, collaborate globally, and deliver infrastructure assets that perform sustainably. Bentley has total annual revenues of $500 million and employs over 3,000 staff worldwide; over $200 million in revenue is derived from Bentley's GI business.

Bentley's flagship software product, MicroStation, supports a number of applications and serves the needs of a range of communities. Bentley Map is a system designed to address the needs of organizations that map, plan, design, build, and operate the world's infrastructure. It enhances the underlying MicroStation platform with capabilities for geospatial data creation, maintenance, and analysis. With Bentley Map, users can integrate data from a wide variety of sources into engineering and mapping workflows. Because Bentley Map is tightly integrated with MicroStation, it allows simultaneous manipulation of raster and vector data, which are standard features of the core platform. An interoperability environment makes it easy to work with many spatial data formats. Bentley's GI system can be implemented with any database connection supported by MicroStation (e.g., Oracle Spatial or ArcGIS Server).

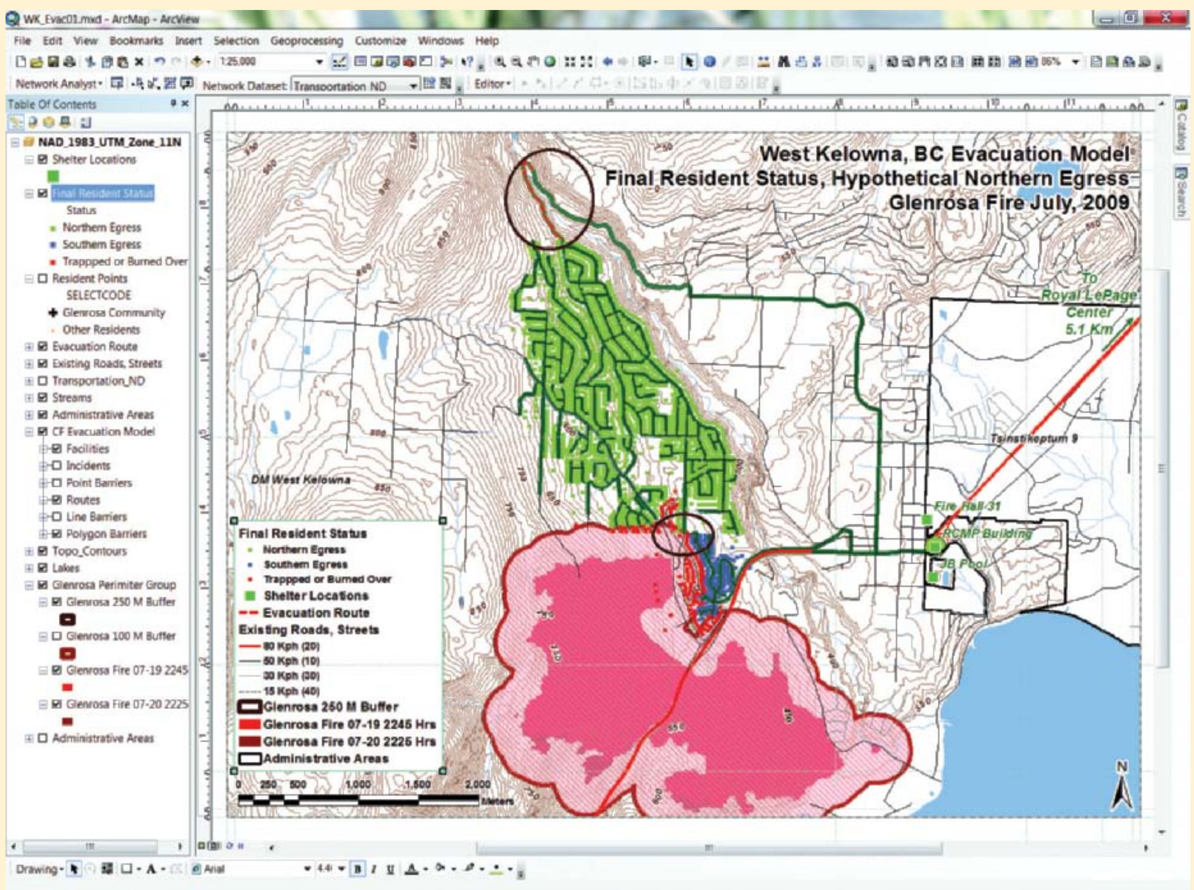
6.5.3 Esri

Esri is a privately held company founded in 1969 by Jack and Laura Dangermond. Headquartered in Redlands, California, Esri employs over 6,000 people worldwide and has annual revenues of over $1 billion. Today it serves more than 300,000 organizations and more than 2 million users. Esri focuses solely on the GI systems market, primarily as a software product company, but it also generates about a sixth of its revenue from project work such as advising clients on how to implement GI systems. Esri started building commercial software products in the late 1970s. Today Esri's product strategy is centered on an integrated family of products called ArcGIS (Box 6.3). The ArcGIS platform is aimed at both end users

Application Box 6.3

Desktop GI System: Esri ArcGIS Desktop

ArcGIS for Desktop (Figure 6.9) is the desktop part of Esri's ArcGIS platform, an integrated suite of products for all key IT environments (desktop, server, Cloud, and mobile). It supports the full range of GI system functions including data collection and import; editing, restructuring, and transformation; display; query; and analysis. A collection of analytical extensions is available for 3-D analysis, network routing, geostatistics, and spatial (raster) analysis, among others. ArcGIS for Desktop is available in three functionality levels: Basic, Standard, and Advanced. Its strengths include a comprehensive portfolio of capabilities, high-quality cartography and



Source: Esri

Figure 6.9 Screenshot of Esri ArcGIS for Desktop.

display tools, analysis functions, extensive customization options, strong data management (using ArcGIS Server), and a vast array of third-party tools and interfaces.

ArcGIS for Desktop comprises a series of integrated menu-driven, end-user applications: ArcMap—a map-centric application supporting integrated editing and viewing; ArcCatalog—a data-centric application for browsing and managing geographic data in files and databases; and ArcToolbox—a tool-oriented application for performing geoprocessing tasks such as proximity analysis, map overlay, and data conversion. ArcGIS is customizable using any Microsoft .Net-compliant programming language such as Visual C^{++} and C#. The software is also notable because of the ability to store and manage all data (geographic and attribute) in standard commercial off-the-shelf DBMS (e.g., DB2, Informix, SQL Server, Oracle, and Postgres).

ArcGIS is the standard against which other products are often compared. It is functionally very rich, but at times can appear complex to new users. It has evolved considerably over the past 30 years, always being reinvented for new technology generations and application requirements.

(especially professional or highly technical users) and technical developers and includes products that run on handheld devices, desktop personal computers (Figure 6.9), servers, and over the Web. Most recently, the company has released an advanced Cloud GI system called ArcGIS Online. ArcGIS Explorer is Esri's virtual globe and is not unlike Google Earth and Microsoft Bing Maps.

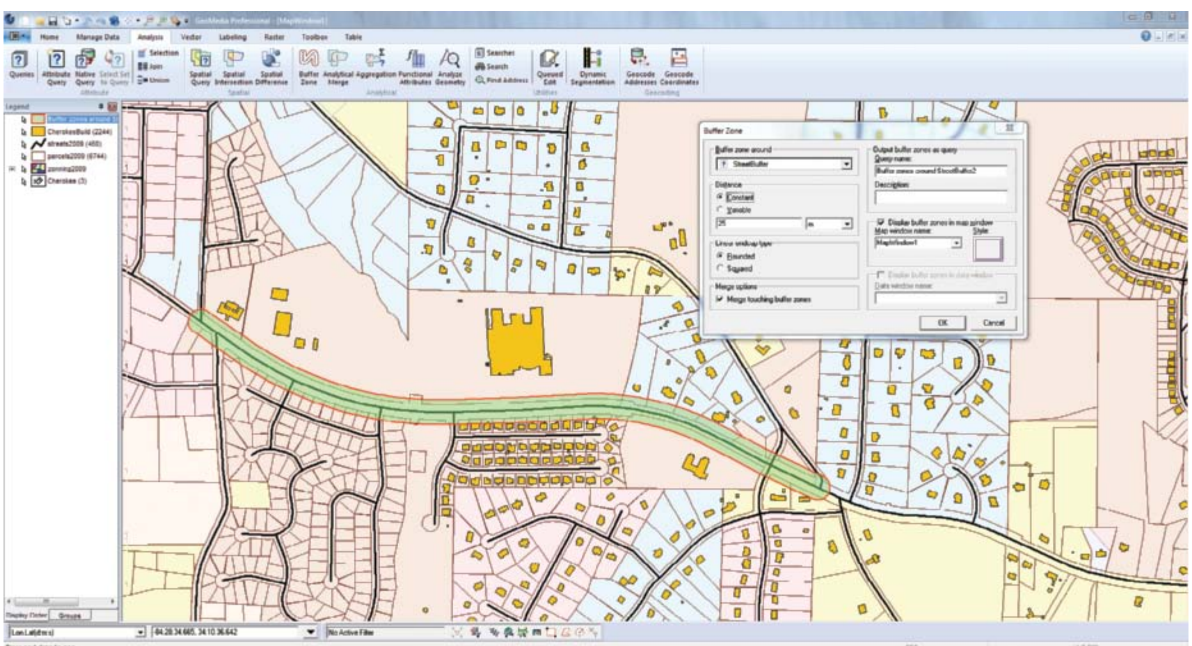
Esri is the classic high-end GI system vendor. It has a wide range of mainstream products covering all the main technical and industry markets. Esri is a technically led geographic company focused squarely on the needs of hard-core users; in recent years it has started to address the needs of less technical mapping users. As a company, Esri is also very much concerned with the practical applications of GI, producing everything from data to data models and Web mapping services.

6.5.4 Intergraph

Like Esri, Intergraph was also founded in 1969 as a private company. The initial focus from their Huntsville, Alabama, offices was the development of computer graphics systems. After going public in 1981, Intergraph grew rapidly and diversified into a range of graphics areas including CAD and mapping software, consulting services, and hardware. Today Intergraph is part of Hexagon, a global provider of design, measurement, and visualization technologies.

Intergraph's geospatial products are able to work with content collected from a variety of sources. The portfolio includes the desktop GI system (Figure 6.10),

Figure 6.10 Screenshot of Intergraph GeoMedia desktop system.



Source: Intergraph

Application Box 6.4

Desktop GI System: Intergraph GeoMedia

GeoMedia is an archetypal example of a mainstream commercial desktop product (Figure 6.10). First released in the late 1990s, it was created from the ground up to run on the Windows desktop operating system. Like other products in the desktop category, it is primarily designed with the end user in mind (rather than the technical user). It has a Windows-based graphical user interface and many tools for editing, querying, mapping, and spatial analysis. Data can be stored in proprietary GeoMedia files or in a DBMS such as Oracle, Microsoft Access, or SQL Server. GeoMedia enables data from multiple disparate databases to be brought into a single environment for viewing, analysis, and presentation. The data are read and translated on the fly directly into memory. GeoMedia provides access to all major geospatial/CAD data file formats and to industry-standard relational databases.

GeoMedia is built as a collection of software components. These underlying objects are exposed to developers who can customize the software using a programming language such as Visual Basic or C#. It offers a suite of analysis tools, including attribute and spatial query, buffer zones, spatial overlays, and thematic analysis. The product's layout tools provide the flexibility to design a range of types of maps that can be distributed on the Web, printed, or exported as files.

GeoMedia is available in three functionality levels: Essentials, Advantage, and Professional. Other members of the product family include GeoMedia Viewer (a free data viewer) and Geo-Media WebMap (for Internet publishing).

All in all, the GeoMedia product family offers a wide range of capabilities for core GI activities in the mainstream markets of government, education, and private companies. It is a modern and integrated product line with strengths in the areas of data access and user productivity.

remote sensing, and photogrammetry software, as well as the synthesis of these technologies in server-based products specializing in data management, spatial data infrastructure, workflow optimization and Web editing, and Web mapping.

From a GI perspective the principal product family is GeoMedia, which spans the desktop and Web-based GI system markets across a range of application domains. Box 6.4 describes Intergraph GeoMedia.

6.6 Types of GI Systems

Over 100 commercial software products claim to have mapping and GI system capabilities. The main categories of generic GI software that dominate today are desktop, Web mapping, server, virtual globe, developer, and mobile. The distinction between the Web mapping, server, and virtual globe categories is sometimes blurred, not least because increasingly they are being deployed in Cloud computing environments. Web mapping systems are those that integrate software and data to create a unified online mapping service; servers typically have a wider range of more advanced functionality and can work with multiple data sources; and virtual globes provide 3-D visualization, query, and some analysis capabilities. In this section, the categories will be discussed followed by a brief summary of other types of software. Reviews of currently popular software packages can be found in the various GI magazines on the Web.

6.6.1 Desktop Systems

Since the mid-1990s, desktop GI systems have been the mainstay of the majority of operational implementations. Desktop software owes its origins to the personal computer and the Microsoft Windows operating system. Desktop software provides personal productivity tools for a wide variety of users across a broad cross section of industries. PCs are widely available, are relatively inexpensive, and offer a large collection of user-oriented tools, including databases, word processors, and spreadsheets. The desktop software category includes a range of options from simple viewers (such as Esri ArcGIS ArcReader, Intergraph GeoMedia Viewer, Pitney Bowes MapInfo ProViewer, as well as a growing number of open-source products) to desktop mapping and GI system software (such as Autodesk AutoCAD Map 3D, Clark Labs Idrisi, Esri ArcGIs; see Box 6.3, Figure 6.9), open-source GRASS, Intergraph GeoMedia (see Box 6.4, Figure 6.10), the Manifold System (Figure 6.11), Pitney Bowes MapInfo Professional, Supermap, Smallworld Spatial Intelligence, and open-source Quantum GIS (see Figure 6.12).

Source: Manifold

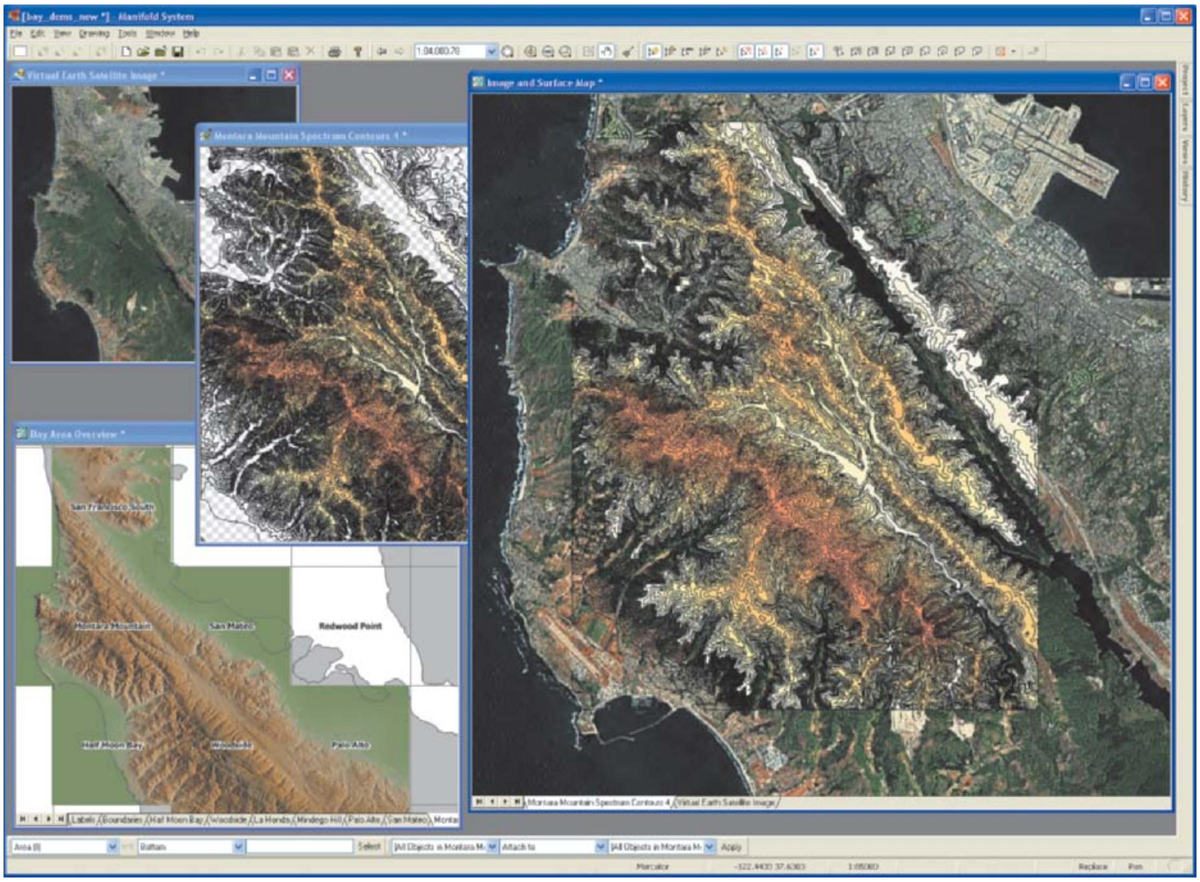
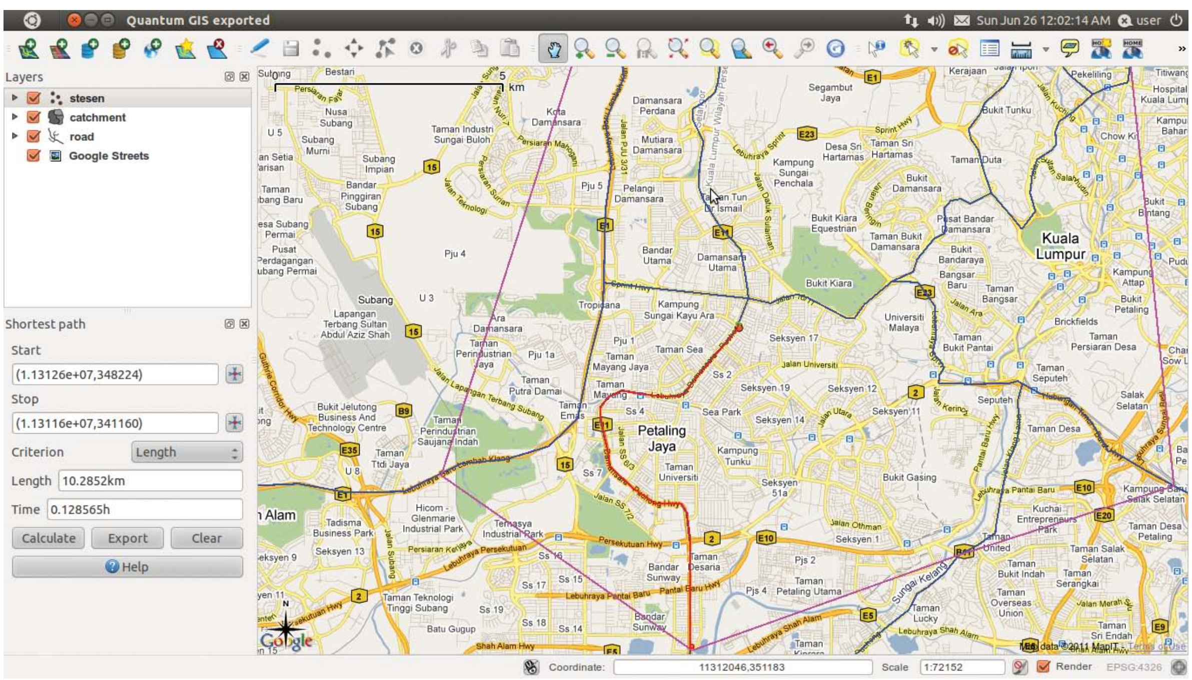


Figure 6.11 Screenshot of Manifold desktop system.

Figure 6.12 Screenshot of Quantum GIS desktop system.



Source: Quantum GIS. www.qgismalaysia.blogspot.com

Desktop GI systems are the mainstream workhorses of today.

In the late 1990s, a number of vendors released free GI viewers that are able to display and query popular file formats. Today, the GI viewer has developed into a significant product subcategory. The rationale behind the development of these products from the commercial vendor point of view is that they help to establish market share and can create de facto standards for specific vendor terminology and data formats. A number of open-source products have also been developed to provide free and easy access to GI and tools. Users often work with viewers on a casual basis and use them in conjunction with more sophisticated software products. Viewers have limited functional capabilities restricted to display, query, and simple mapping. They do not support editing, sophisticated analysis, modeling, or customization.

With their focus on data use rather than data creation and their excellent tools for making maps, reports, and charts, desktop mapping and GI systems represent most people's experience of high-end systems today. Many of the successful systems have adopted Microsoft standards for interoperability and user-interface style (although some open-source systems are based on Java). Users often see a desktop mapping package and a GI system as simply a tool to enable them to do their full-time job faster, more easily, or more cheaply. Desktop mapping and GI system users work in planning, engineering, teaching, the army, marketing, and other similar professions; they are often not advanced technical users. Desktop GI system prices typically range from $400 to $1,500 (these and other prices mentioned later typically have discounts for multiple purchases).

The most advanced and functionally rich desktop software packages include data collection and editing, database administration, advanced geoprocessing and analysis, and other specialist tools. Such professional software packages offer a superset of the functionality of the systems in other classes. The people who use these systems typically are technically literate and think of themselves as GI professionals (career GI staff); they have degrees and, in many cases, advanced degrees in GI science or related disciplines. Prices for professional GI systems are typically in the $5,000–$15,000 range per user.

Professional GI systems are high-end and fully functional systems.

6.6.2 Web Mapping Systems

GI systems on the Web have a comparatively long history that goes back to 1993 when researchers at Xerox PARC in California created the first map server that could be accessed over the Internet. Since then several major milestones have been reached in developmental and organizational terms: the popularity of the MapQuest site and subsequent acquisition of the company by AOL for over $1.1 billion; the introduction of major mapping sites by Google, Yahoo!, and Microsoft (among others); and the rise of computer mashups and neogeography (see Section 6.6.3 and Chapter 10), to name but a few.

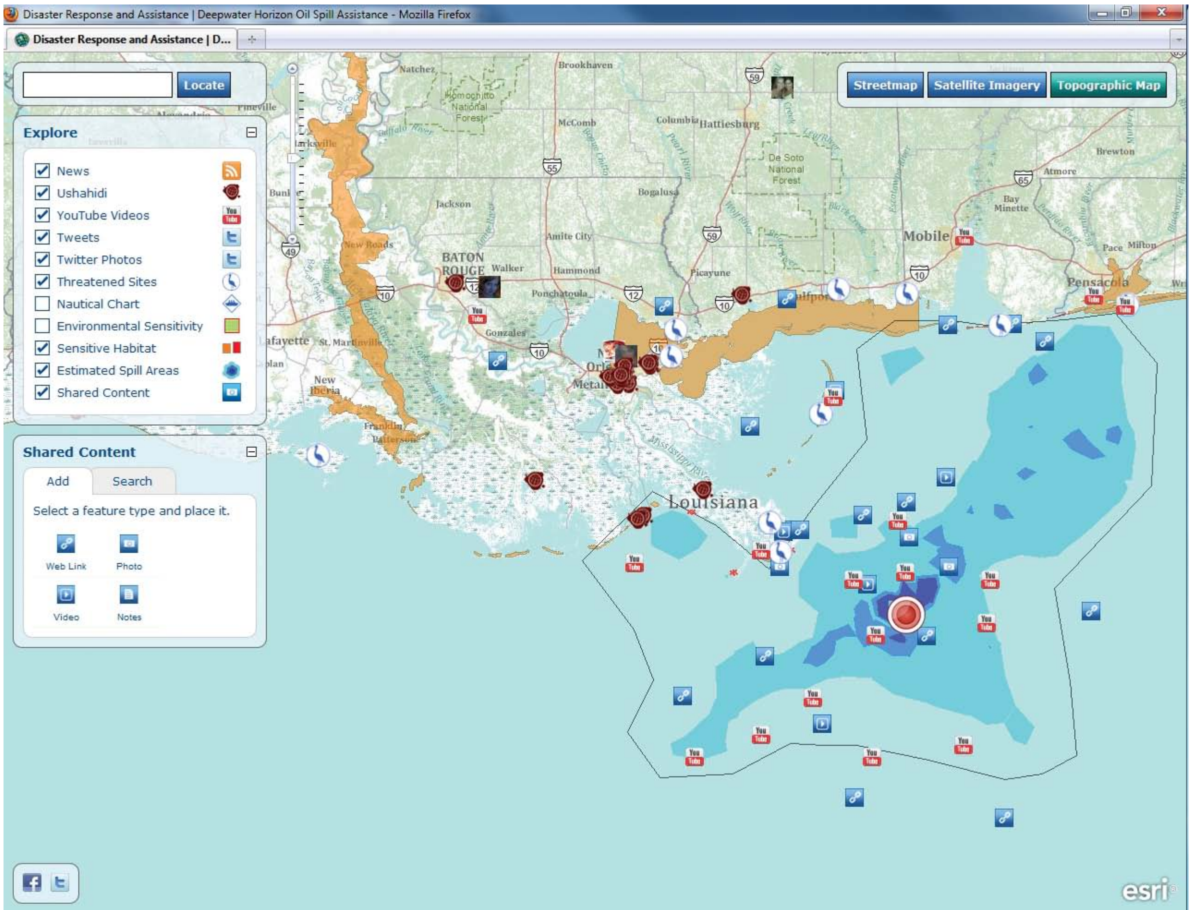
Here the term *Web mapping* is taken to mean integrated Web-accessible software, a 2-D database (3-D base maps are covered later in Section 6.6.4 on virtual globes) comprising one or more base maps, and an associated collection of services. Web access is provided via easily accessible, open interfaces. Typically, image tiles—each containing a fragment of a map—are returned in response to a user request for a map of a given area. Requests can be issued from, and responses processed and visualized in, standard Web browsers and Web-accessible devices (e.g., phones and tablets), and Web mapping sites have proven to be very popular (being consistently rated among the top 10 in terms of Web site traffic). In addition to providing maps, a range of other useful services are normally closely coupled, for example, a gazetteer to find places of interest, multipoint driving directions, a choice of image and street map base maps, and the ability to overlay, or mash up, many other datasets.

Some of the success of Web mapping sites is also due to the fact that they can be easily accessed programmatically via well-defined APIs. The quintessential example here is KML and the associated API for interacting with Google Maps (see Box 6.1). Web APIs have spawned a large number of add-ons and have allowed Web maps to be integrated or mashed up with many other Web services. Figure 6.13 shows multiple Web services overlaid (mashed up) on a topographic base map; see Section 10.4.1 for a further example.

OpenStreetMap is a base-map service collected by volunteers and published on the Web using a copyright-free license. The software used to edit, access, and view the data is open source (including Mapnik for map rendering, JOSM and Potlatch for editing, and PostGIS for data management). In certain cities, OpenStreetMap is now beginning to rival and even exceed the content and coverage of commercial online base-map databases.

6.6.3 Server GI Systems

In simple terms, a server GI system is a system that runs on a computer server and can handle concurrent processing requests from a range of networked



Data courtesy: U.S. Fish & Wildlife Service, National Park Service, LA Dept. of Wildlife & Fisheries, NOAA, Ushahidi, YouTube, and Twitter

Figure 6.13 A situational awareness viewer for an oil spill disaster that mashes up several social media feeds on top of a Web-service base map.

clients. Server GI systems offer a wider range of functions than Web mapping systems, which focus solely on mapping and closely related services and can work with any base map. Server GI system products have the potential for the largest user base and lowest cost per user of all software system types. Stimulated by advances in server hardware and networks, the widespread availability of the Internet, and market demand for greater access to geographic information, GI system vendors have been quick to release server-based products that are accessible over the Web. Examples of server systems include Esri ArcGIS Server, Intergraph GeoMedia Webmap, MapGuide and MapServer (both open source), Pitney Bowes MapInfo MapXtreme, and Smallworld Spatial Application Server. The cost of commercial server products varies from around $5,000 to $25,000, for small to medium-sized systems, to well beyond for large multifunction, multiuser systems. Box 6.5 highlights the capabilities of MapGuide.

Server GI systems are growing in importance as capabilities increase and organizations shift from desktop to network implementations.

The most popular open-source GI system today is probably the MapServer Web mapping system (Figure 6.14). Originally developed at the University of Minnesota, it now has community ownership. The primary purpose of MapServer is to display and interact with dynamic maps over the Internet. It can work with hundreds of raster, vector, and database formats and can be accessed using popular scripting languages and development environments. It supports state-of-the-art functionality such as on-the-fly projections, high-quality map rendering, and map navigation and queries. MapServer has been used quite widely as the core of many spatial data infrastructure projects to render maps and respond to queries for data download.

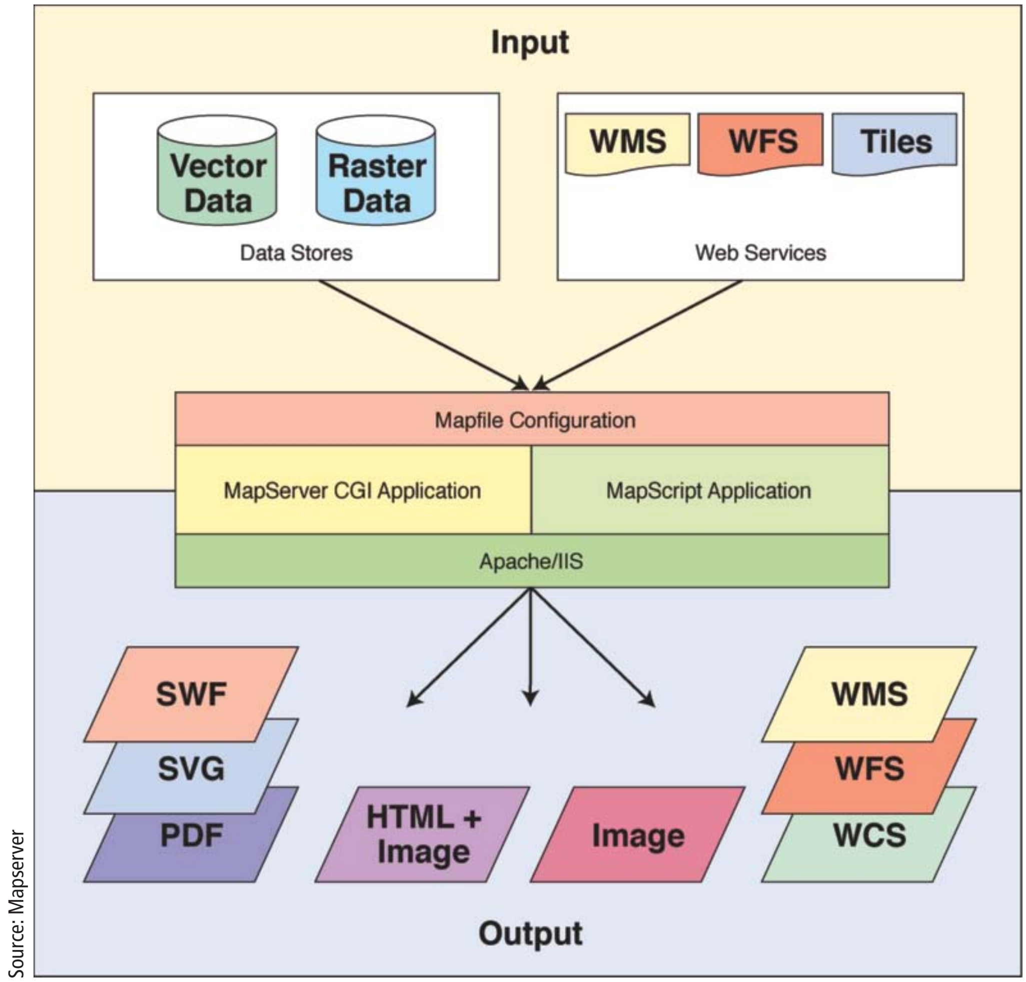


Figure 6.14 Architecture of MapServer. The central boxes are the software components that read data, render maps, and fulfill other client requests. A Web server (open-source Apache or Microsoft IIS) is used to communicate with browser clients.

Application Box 6.5

Server System: MapGuide Open Source

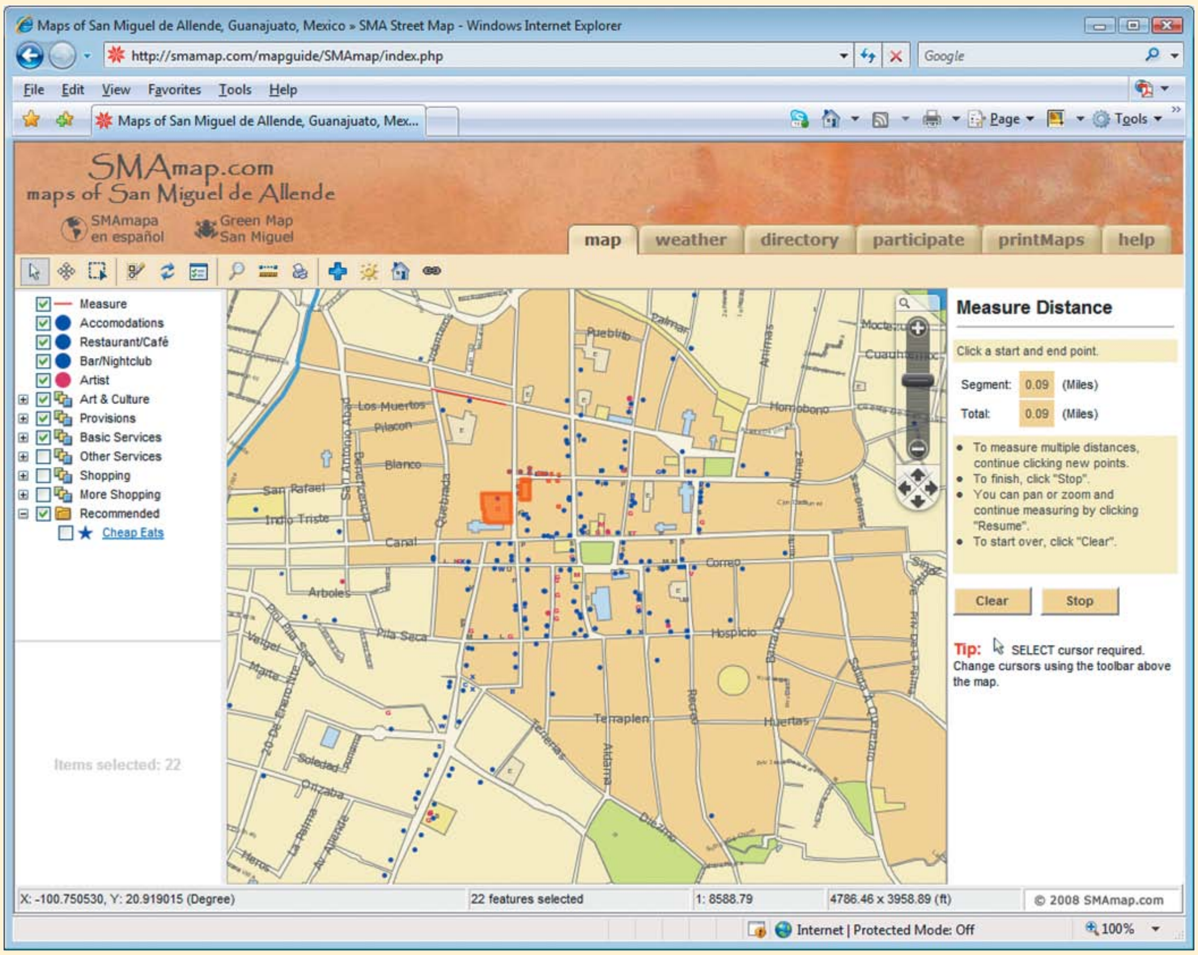
MapGuide Open Source is used to develop and deploy Web mapping applications and Web services. It is licensed under the open-source GNU Lesser General Public License. MapGuide is an important system for the many users who have spent considerable amounts of time and money creating valuable databases and who want to make them available to other users inside or outside their organization. It allows users to leverage their existing GI investment by publishing dynamic, intelligent maps at the point at which they are most valuable—in the field, at the job site, or on the desks of colleagues, clients, and the public.

Figure 6.15 is a screenshot of a MapGuide application that provides tourists and residents an easy way to locate streets and businesses in or around San Miguel de Allende, Guanajuato, Mexico. Using this application, community members are encouraged to include small local businesses that would not otherwise have an Internet presence.

There are three key components of MapGuide: the viewer—a relatively easy-to-use Web application with a browser-style interface; the author—a menu-driven authoring environment used to create and publish a site for client access; and the server—the administrative software that monitors site usage and manages requests from multiple clients and to external databases. MapGuide works directly with Web browsers and servers and uses the HTTP (Internet protocol) for communication.

It makes good use of standard Internet tools like HTML (hypertext markup language) and JavaScript for building client applications. Typical features of MapGuide sites include the display of raster and vector maps, map navigation (pan and zoom), geographic and attribute queries, buffering, report generation, and printing. Like other advanced Web-based server systems, MapGuide has tools for redlining (drawing on maps) and basic editing of geographic objects.

To date, MapGuide has been used most widely in existing mature GI sites that want to publish their data internally or externally and in new sites that want a way to publish dynamic maps quickly to a widely dispersed collection of users (for example, maps showing election results or transportation network status). MapGuide can be used to serve maps (using the OGC WMS protocol) and features (using the OGC WFS protocol). Configured as a mapping service, MapGuide supports a client/server environment. It can retrieve geospatial data from WFS and WMS sites, enabling the use of data from other organizations that share their geospatial data. As a further service, MapGuide allows organizations to share their data, in vector form, with authorized outside organizations.



Source: MapGuide

Figure 6.15 Screenshot of MapGuide Web mapping application.

Initially, server GI systems were nothing more than ports of desktop products, but second-generation systems were subsequently built using a multiuser services-based architecture that allows them to run unattended and to handle many concurrent requests from remote networked users. These systems often focus on display and query applications—making things simple and cost effective—with more advanced applications available at a higher price. Server GI systems are especially adapted for well-defined tasks that need to be performed repeatedly (ad hoc tasks largely remain the preserve of desktop systems, at least for now). Today, it is routinely possible to perform standard operations like making maps, network routing, geocoding, editing, and publishing many types of thematic and topographic data. A number of Web GI services are indexed using geoportals—gateways to GI resources—for example, the U.S. federal government geoportal www.data.gov. These are built using server software. Many Cloud GI systems are underpinned by server software.

A second generation of Web-based server products is becoming increasingly prevalent. They exploit the unique characteristics of the Web and integrate GI technology with Web browsers and servers. Initially, these new systems had limited functionality, but now there is a new breed of true server products that offer complete GI system functionality in a multiuser, Web-based server environment (e.g., ArcGIS Server). These server products have functions for distributed editing, mapping, data management, and spatial analysis, and support state of-the-art customization.

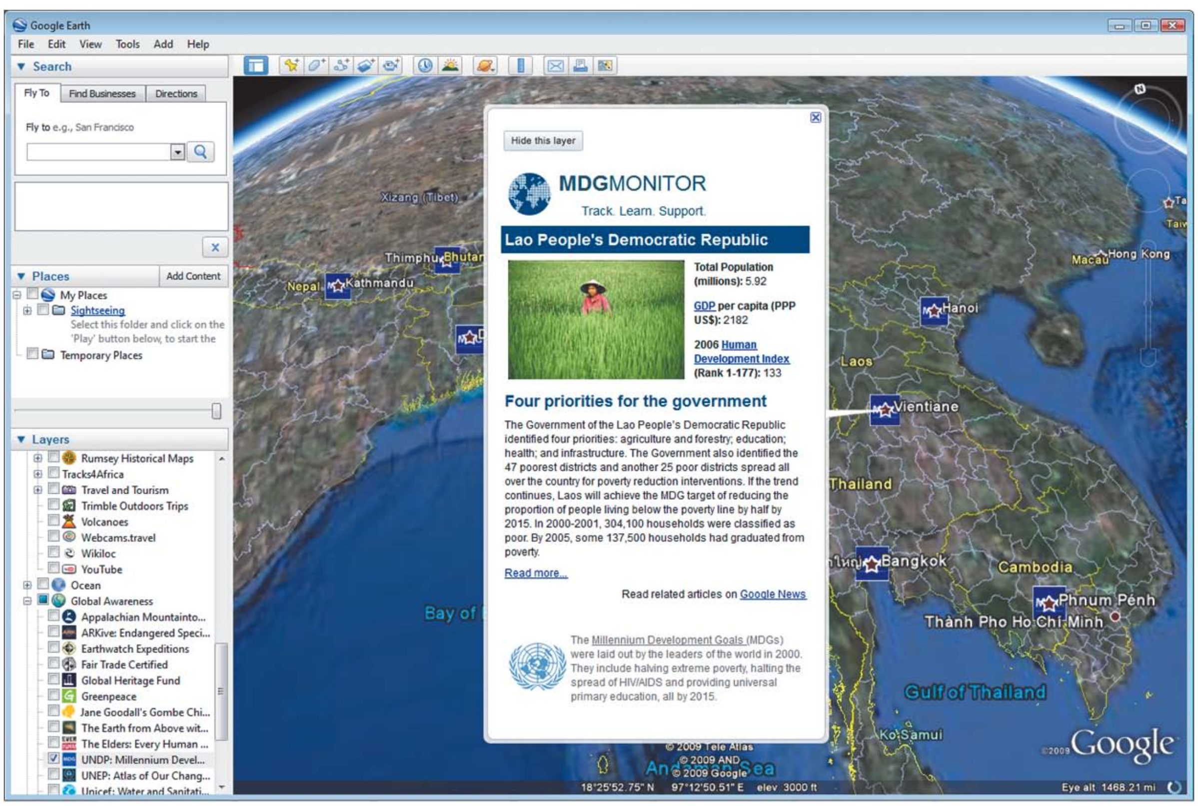
In conclusion, server products are growing in importance. Their cost-effective nature, ability to be centrally managed, and focus on ease of use will help to disseminate GI even more widely and will introduce many new users to the field. Server systems are the heart of many Cloud GI implementations.

6.6.4 Virtual Globes

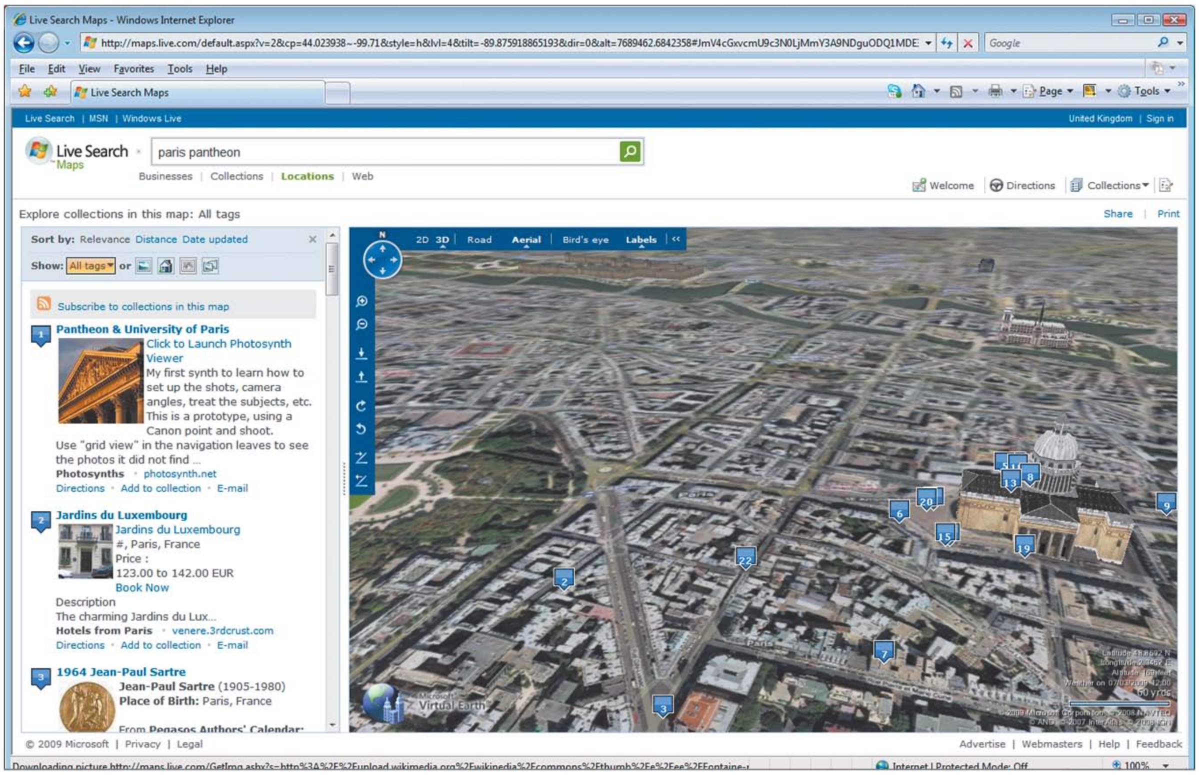
One of the most exciting recent developments in the field has been the advent of virtual globes—3-D Web services hosted on Web-based systems that publish global 3-D databases and associated services for use over the Web. Virtual globes allow users to visualize geographic information on top of 3-D global base maps. Users can fly over a comparatively fine-resolution virtual globe with thematic data overlaid. Google Earth, released in 2005, set the standard for user interaction and the quality and quantity of data and established in a matter of months an engaged user community. Figure 6.16 shows information about the Millennium Development Goals (MDGs) for the Lao People's Democratic Republic as represented in Google Earth. Since then, a number of other vendors have released comparable globes, not least of which is Microsoft Bing Maps 3D. Figure 6.17 shows the center of Paris in the Bing Maps 3D mode.

Virtual globes have gained considerable traction in the wider community because, comparatively speaking, they are low cost (basic versions are free), they have high-quality image and vector databases for the Earth (and other planets), it is simple to overlay user data on a globe, and they are easy to use.

Figure 6.16 Screenshot of Google Earth (earth.google.com).



Source: Google, map data copyright Google, AND, TeleAtlas



Source: Microsoft Bing

Figure 6.17 Screenshot Microsoft Virtual Earth (www.bing.com/maps).

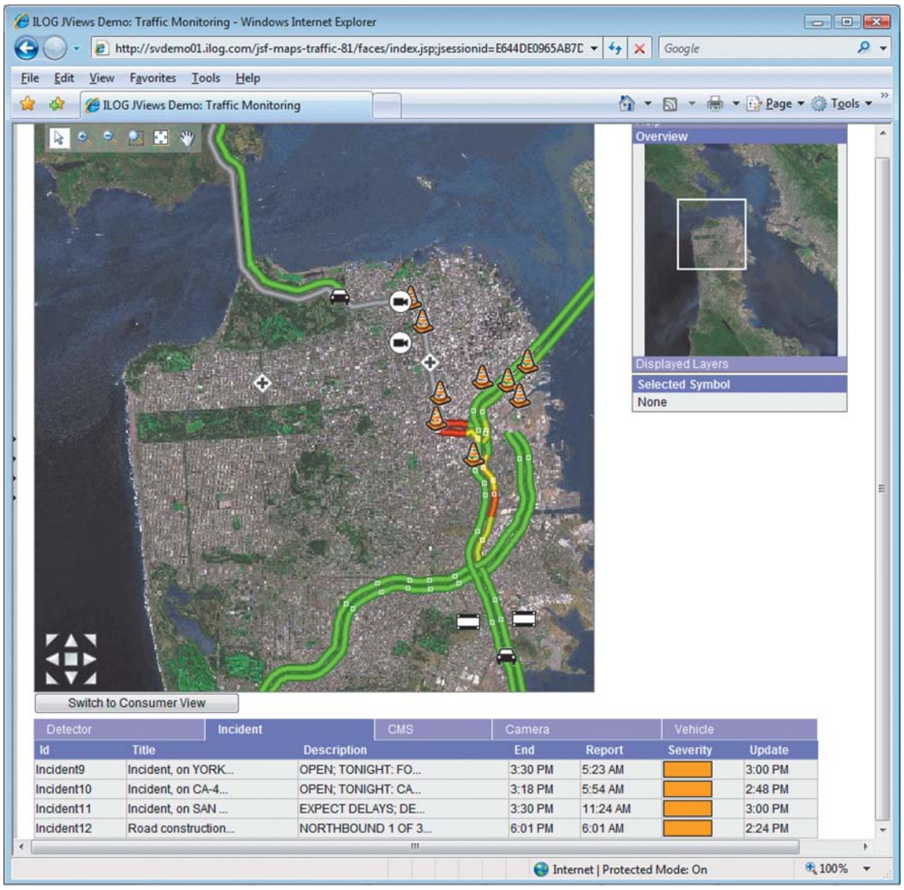
They have stimulated in no small measure two whole new subfields called neogeography (see Section 1.5.5) and volunteered geographic information (VGI; see Sections 1.5.6 and 10.2). Neogeography is the "new" geography that among other things includes the overlay or mashing up of two or more sources of geographic information (for example, webcams from Caltrans [California Department of Transportation] on top of a Yahoo! base map). VGI focuses on the fact that humans are acting as sensors and are building and publishing content from the ground up, often using virtual globes as base maps. The nonauthoritative and sometimes transient and dynamic nature of this information provides new geographic challenges and opportunities. These issues are explored in more depth in Chapter 10.

6.6.5 Developer GI Systems

With the advent of component-based software development (see Section 6.4), a number of collections of software components oriented toward the needs of developers have been released. These are really tool kits of GI system functions (components) that a reasonably knowledgeable programmer can use to build a specific-purpose application. They are of interest to developers because such components can be used to create highly customized and optimized applications that can either stand alone or can be embedded within other systems. Typically, component packages offer strong display and query capabilities, but only limited editing and analysis tools, mainly because there is greatest demand for products with such data exploration and visualization functionality. However, this is beginning to change as products mature. In the past few years there has been a surge in Web services that can be customized and embedded using published APIs (see Box 6.1).

Developer products are collections of components used by developers to create focused GI applications.

Esri ArcGIS Engine is a good example of a desktop development product that can be used in .Net, and Java environments. Java-based tool kits include ObjectFX SpatialFX and IBM ILOG JViews Maps (see Figure 6.18). The typical cost for a commercial developer product is $1,000–5,000 for the developer kit and $50–500 per deployed application. The people who use deployed applications may not even realize that they are using a GI system because often the run-time deployment is embedded in other applications (e.g., customer care systems, routing systems, or interactive atlases).



Source: IBM ILOG

Figure 6.18 Screenshot of a Web application written using the IBM ILOG JViews Map component tool kit.

There are a number of open-source developer toolkits of which GeoTools and GDAL are two of the most widely used. GeoTools is a Java-based code library that provides OGC standards-compliant methods for the manipulation of geospatial data. It has been used in a number of GI systems including GeoServer and is a key part of the GeoAPI set of tools. GDAL/ODR is a software library for reading and writing raster and vector data formats. It is widely used in commercial and open-source products.

6.6.6 Mobile GI Systems

As hardware design and miniaturization have progressed dramatically over the past few years, so it has become possible to develop software for mobile and personal use on handheld systems. The development of low-cost, lightweight location positioning technologies (primarily based on the Global Positioning System; see Section 4.9) and wireless networking has further stimulated this market. With capabilities similar to the desktop systems of just a few years ago, mobile smartphones, tablets, and specialist devices can support many display, query, and simple analytical applications, even on displays as small as 320 by 240 pixels (although today much finer-resolution screens are available). An interesting characteristic of these systems is that all programs and data are held in local memory because of the lack of a hard disk. This provides fast access, but because of the cost of memory compared to disk systems, designers have had to develop compact data storage structures. Esri's ArcPad product is one of a number of products in this space (Figure 6.19). It specializes in field data collection and mobile mapping. When linked to Esri ArcGIS Server, it can act as an enterprise field client.

Mobile GI systems are lightweight and designed for handheld field use.

One of the most innovative areas at present is the development of software for smartphones and related devices. Despite their compact size, they can deal with comparatively large amounts of data (16 GB and more) and handle surprisingly sophisticated

Source: Esri

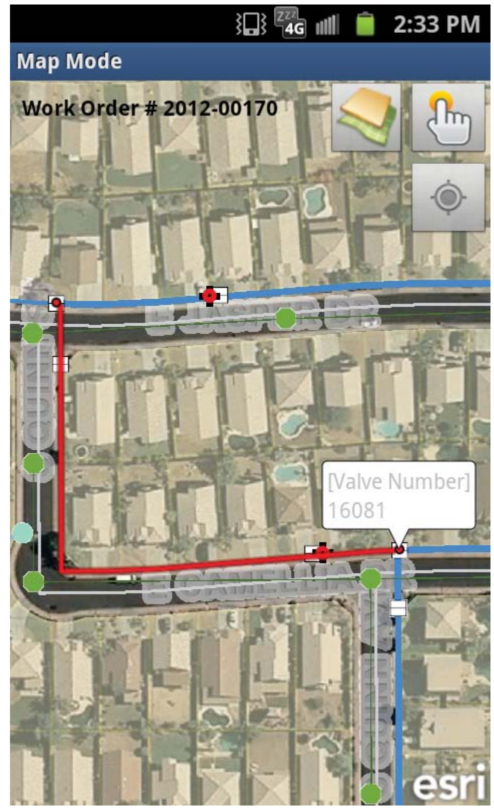
Figure 6.19 Esri ArcPad running on a rugged handheld field portable device.

applications. The systems usually operate in a mixed connected/disconnected environment and so can make active use of data and software applications held on a server (see earlier discussion of server GI systems in Section 6.6.3) and accessed over a wireless telephone or WiFi network. Many applications can run on smartphones, tablets, and other specialist devices and can deliver highly productive specific-purpose GI applications (Figure 6.20).

6.6.7 Other types of GI Software

The previous section focused on mainstream software from the major commercial vendors. There are many other types of commercial and noncommercial software that provide valuable GI capabilities. This section briefly reviews some of the main types.

Raster-based GI systems, as the name suggests, focus primarily on raster (image) data and raster analysis. Chapters 2, 3, and 7 provide a discussion of the principles and techniques associated with raster and other data models, whereas Chapters 13 and 14 review their specific capabilities. Just as many vector-based systems have raster analysis capabilities (for example, Esri ArcGIS has Spatial Analyst, and Intergraph GeoMedia has Image and Grid), in recent years raster systems have added vector capabilities (for example, ERDAS Imagine and Clark Labs Idrisi now have vector capabilities built in; see Figure 6.21). As a result, the distinction between raster-based and other system categories is becoming increasingly blurred. The users of raster-based systems are primarily interested in working with imagery and undertaking spatial analysis and modeling activities. The prices for raster-based systems range from $500 to $10,000.



Source: Esri

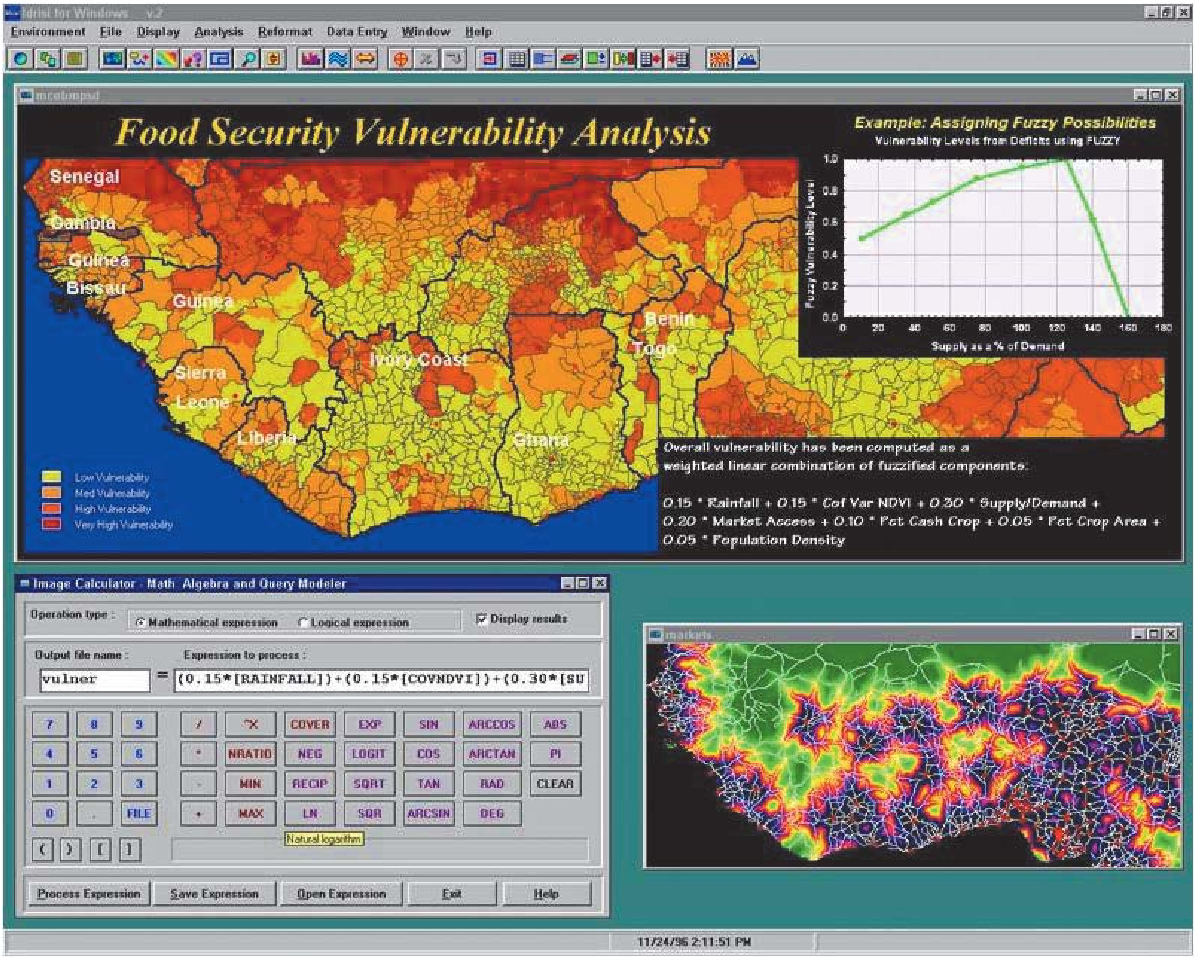
Figure 6.20 A utility work order application that runs on a smartphone device.

CAD (or computer-aided design)-based systems started life as CAD packages and then had GI functions added. Typically, a CAD system is supplemented with database, spatial analysis, and cartography capabilities. Not surprisingly, these systems appeal mainly to users whose primary focus is in typical CAD application areas such as architecture, engineering, and construction, but who also want to use GI and geographic analysis in their projects. These systems are often used in data collection and mapping applications. The best-known examples of CAD-based systems are Autodesk Map 3D and Bentley Map (see Section 6.5). CAD-based GI systems typically cost from $3,000 to $5,000.

To assist in managing data in standard DBMS, some vendors—notably IBM, Microsoft, and Oracle—have developed technology to extend their DBMS servers so that they are able to store and process GI efficiently. Although not strictly GI systems in their own right (because of the absence of editing, mapping, and analysis tools), they are included here for completeness. Box 9.1 provides an overview of Oracle's Spatial DBMS extension.

Some noteworthy examples of open-source systems not already mentioned in earlier sections include GRASS (desktop), GeoDA for spatial analysis and visualization, gvSIG for desktop mapping, and PostGIS for storing data in a DBMS.

The Open Geospatial Consortium, although not itself a software vendor or software-development organization, has played a very important role in the evolution of GI software. OGC's role has been to encourage the development of standards that facilitate the sharing of GI and the interaction of geographic software. Perhaps the most significant progress has been in the area of interoperability of Web services, and a series of OGC standards now allow images, features, coverages, and metadata to be integrated together over the Web.



Source: Clark Labs

Figure 6.21 Screenshot of Clark Labs Idrisi desktop system.

6.7 Conclusion

Software is a fundamental and critical part of any operational GI system. The software employed in a project has a controlling impact on the type of studies that can be undertaken and the results that can be obtained. There are also far-reaching implications for user productivity and project costs. Today, there are many types of software to choose from and a number of ways to configure implementations. One of the exciting and at times unnerving characteristics of GI software is its very rapid rate of development, not least in the areas of Web and open-source systems. This trend seems set to continue as software developers push ahead with significant research and development efforts. The following chapters explore in more detail the functionality of GI software and how it can be applied in real-world contexts.

Questions for Further Study

1. Design a GI system architecture that 25 users in three cities could access to create an inventory of recreation facilities.
2. Discuss the role of each of the three tiers of software architecture in an enterprise GI implementation.
3. With reference to a large organization that is familiar to you, describe the ways in which its staff might use a GI system, and evaluate the different types of systems that might be implemented to fulfill these needs.
4. Go to the Web sites of the main GI system vendors, and compare their product strategies with those of open-source GI products such as OpenGeo. In what ways are they different?
   - Autodesk: www.autodesk.com
   - Bentley: www.bentley.com
   - Esri: www.esri.com
   - Intergraph: www.intergraph.com
   - OpenGeo: opengeo.org

Further Reading

Coleman, D. J. 2005. GIS in networked environments. In Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. (eds.). *Geographical Information Systems: Principles, Techniques, Management and Applications* (abridged edition). Hoboken, NJ: Wiley, 317–329.

Fu, P. and Sun, J. 2011. *Web GIS: Principles and Applications*. Redlands, Esri Press.

Maguire, D. J. 2005. GIS customization. In Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. (eds.). *Geographical Information Systems: Principles, Techniques, Management and Applications* (abridged edition). Hoboken, NJ: Wiley, 359–369.

Tsou, M.-H., and Smith, J. 2011 *Free and open source software for education*. National Geospatial Technology Centre of Excellence geoinfo.sdsu.edu/hightech/WhitePaper/tsou_free-GIS-for-educators-whitepaper.pdf