INT. J. GEOGRAPHICAL INFORMATION SCIENCE, 1999, VOL. 13, NO. 4, 429–438

Technical Communication

# Interactive visualisation of geographical objects on the Internet*

JAAKKO KÄHKÖNEN, LASSI LEHTO, TIINA KILPELÄINEN and TAPANI SARJAKOSKI

Finnish Geodetic Institute, Department of Cartography and Geoinformatics, Geodeetinrinne 2, P.O. Box 15, FIN-02431 Masala, Finland

email: {jaakko.kahkonen, lassi.lehto, tiina.kilpelainen, tapani.sarjakoski}@fgi.fi

**Abstract.** The potential of the World Wide Web as an information distribution network has increased. The expansion of the Internet provides data suppliers with a completely new means for disseminating geospatial information in a visual and interactive manner to the general public. Recent developments indicate that the Web is gradually being transformed from a distributed document system towards a distributed application framework. Several new object-oriented techniques have been introduced as a solution for dynamic processing of information on the Web. The present paper illustrates a research project carried out at the Finnish Geodetic Institute to develop a viewer application (NetGIS) aimed at giving an ordinary Web surfer an opportunity to browse and visualise geographical information (GI) in the form of spatial objects. This new object-based approach to the delivery of GI on the Web enables interactive querying of the properties of an individual geographical object. The application is implemented in Java programming language and the data transfer is based on use of the CORBA standard and on OpenGIS Simple Features Specification.

## 1. Introduction

The Internet, originally a text-based communication tool, is developing towards a multimedia communication network. A global hypermedia infrastructure has been realized in the form of the World Wide Web (Web), based on new standards and existing protocols (W3C 1998). Development of the Web platform has been accelerated by the popularity of the easy-to-use graphical user interface provided by Web browsers such as Netscape Navigator® or Microsoft Internet Explorer®; thus, the importance of the Internet as an information distribution network has increased.

### 1.1. *Delivery of geographical information (GI) on the Web*

The rapid expansion of the Internet provides also spatial data suppliers with a completely new means for disseminating GI to the general public. At the same time digital maps can be seen as an interface to the vast amount of information contained in public networks (McKee 1996).

The use of static maps in raster form, displayed in a Web browser not providing any analysis functions, severely limits facilities available to the end user. Some new

*Web supplement located at: http://www.fgi.fi/osastot/karto/projektit/netgis/vis_98/ngsh.html

*International Journal of Geographical Information Science*
ISSN 1365-8816 print/ISSN 1362-3087 online 
http://www.tandf.co.uk/JNLS/gis.htm
http://www.taylorandfrancis.com/JNLS/gis.htm

technologies have emerged as possible alternatives to the plain HTML (HyperText Markup Language) solution, to implement more intelligent, interactive client applications. These clients are also capable of handling vector-based GI, possibly augmented with visual or aural extensions, such as video and animation with sound effects, i.e. versatile multimedia data. The new technologies open to the wide Internet audience give access to spatial information in the form of intelligent objects equipped with properties and behaviour. The technologies in the spotlight here are the Java™ programming language and Common Object Request Broker Architecture (CORBA™).

### 1.2. *Need for interoperability*

On the Internet a number of heterogeneous applications and computers are connected and should be able to communicate together. Finding usable solutions for interoperability is one of the demanding challenges to data producers and application developers. End users are exhausted by endless software updating, data transferring and conversion between different GIS applications and platforms, before the data is available to users who need GI to be analysed and visualised. A myriad of different file formats has been developed over the years and numerous conversion packages exist to process information from one format to another. The current Web mapping products provided by traditional GIS vendors are also based on proprietary solutions—a fact effectively preventing multi-server browsing of GI.

The situation is about to change. The network-centric processing of information, based on the principles of distributed computing, thin-client technology and platform-independent executables, creates a computing architecture that supports real-time interoperability. A distributed computing model, such as CORBA, combined with dynamically downloadable Java-based client applications, fits well into the Web environment. On top of these general information technology foundations, domain-specific interface standards can be developed. A significant initiative in this respect is the Open GIS Consortium's (OGC 1998) recent OpenGIS™ Simple Features Specification (SFS). In the present paper the emphasis is on the SFS version for the CORBA environment.

Here we present a research project carried out at the Finnish Geodetic Institute to develop a viewer application, capable of presenting an ordinary Web surfer with GI in the form of intelligent spatial objects (Lehto *et al.* 1997). In the following a review is given of the main technologies used in the development of the application, i.e. Java, CORBA and the SFS. After that, the functionality and technical features of NetGIS, the developed Java-based spatial Web browser prototype, are described.

## 2. Platform-independent technologies

Since the possibilities available for the exchange of GI between systems have been based on tedious transfer file-based processes, it is rational to look for platform-independent techniques to make it easier to combine various data sources and to visualise them separately, or combined for various purposes.

### 2.1. *Java*™

Developments in information technology clearly indicate that the Web is gradually being transformed from a distributed document system towards a distributed application framework. Several new object-oriented techniques have been introduced as a solution for dynamic processing of information on the Web. One of these new technologies is the Java programming language (Arnold and Gosling 1996).

The Java language developed by JavaSoft, a Sun Microsystems company

(JavaSoft 1998), is rapidly becoming the predominant programming tool for developing small applications, called applets, to be embedded as active components in an HTML document. The applet code is downloaded dynamically to the Web browser the moment the corresponding HTML document is requested. Java code is comparable to other high-level programming languages but differs from them in compilation strategy. The Java source code is compiled into a platform-independent byte code. The byte code is interpreted at run time by Java Virtual Machine (JVM). Inclusion of JVM in the most popular Web browsers created a wide base of Java-enabled platforms. The complete Java platform consists of the programming language, JVM and the ever increasing number of Application Programming Interfaces (API), software libraries for various areas of computing.

Dynamic downloading of the platform-independent Java code provides an elegant solution for implementing a GI browser, suitable even for a casual end user. Since the code needed to work with geographical objects is downloaded to the local CPU together with the dataset, possession of an adequate Internet connection and a Java-enabled Web browser remains the only end-user prerequisite for processing object-based GI. The platform-independent nature of the Java byte code relieves data providers of the burden of attempting to provide a separate software version for each platform the end user may be working on. Transfer of executable code to the local CPU enables the client application to be programmed at an arbitrary level of sophistication. Duties can thus be divided between the server and the client as considered appropriate by the system designer.

The Java language has been designed from the outset for a networked environment. Support for network communications, readily available in the Java core libraries, explains the rapid adoption of Java as a predominant Web language. Network support includes such central Internet and Web concepts as TCP/IP socket (Transmissions Control Protocol/Internet Protocol), UDP socket (User Datagram Protocol) and URL (Uniform Resource Locator). A distributed computing mechanism for pure Java solutions, referred to as Remote Method Invocation (RMI), has been developed on top of these basic transport protocols.

RMI is the realization in Java language of the remote procedure call (RPC) mechanism found in other programming languages. The basic idea in RPC is to start procedures defined in a given program remotely, i.e. from another computer across the network. The parameters for and return values from a remote procedure are typically objects that are transferred through the network by means of the Java serialization technique.

In addition to easy implementation of client-server applications, RMI also facilitates the design of more complicated configurations. In the RMI environment each software entity is able to act as a server or a client, as considered appropriate by the designer. One widely discussed approach is the three-tiered programming model in which user interface, application logic and database each form a separate entity in a distributed environment (Zhang and Lin 1996). Unfortunately, RMI enables communication only between Java objects. A more general approach is clearly needed. Java has recently been integrated more closely with CORBA, a standard for distributed computing in heterogeneous environments.

### 2.2 *Common Object Request Broker Architecture (CORBA)*

Existing Web protocols do not give the best possible support for managing distributed applications. Consequently, CORBA is about to emerge as a new standard on the Web. CORBA is an object-oriented standard for distributed computing in a heterogeneous environment and is neutral with respect to computing hardware, operating systems and even programming language. Although the CORBA standard has been under development for nearly 10 years, it has only recently found a practical application area: the heterogeneous global Web environment.

A central part of the CORBA standard is the object bus by which the participating applications can communicate with each other (OMG 1997). Interaction between incompatible applications is based on well-defined interfaces that conceal the internal details of one application from the others. The CORBA standard contains a specific language for describing the interfaces (Interface Definition Language, IDL). A concrete CORBA interface can be an object-oriented software module that wraps an existing application and maps the internal data types of the application to the concepts of the defined interface. Communication between applications is mediated by Object Request Brokers (ORB), software modules that transfer messages between each other according to a defined CORBA protocol, the General Inter-ORB Protocol (GIOP). The CORBA object bus delivers remote interfaces (stubs) to the client platform, making server objects look like local counterparts from the viewpoint of the client application. JavaSoft has included the support for CORBA in its Java libraries (Java IDL), and several Java-based CORBA ORB modules have appeared on the market.

Java and CORBA, together with commonly agreed domain-specific standards, form a good basis for designing distributed interoperable GISs for the Web environment (Lehto 1998). Real implementation for the GIS industry needs to be based on a comprehensive standard, such as the OGC SFS (Vckovski 1998; p. 203).

### 2.3. *Open GIS Consortium's (OGC) Simple Features Specification (SFS)*

SFS is an interface definition of how to model GI. It was developed by the OGC, an industry group of over 120 organisations, and has been accepted by all of the major GIS vendors (OGC 1997). The implementation specification for the CORBA platform (the SFS for CORBA) will most likely be used in the Web environment in the near future. It consists of two main parts: the Feature Object Model and the Geometry Object Model. The specification is essentially based on the object-oriented programming paradigm—a piece of GI is represented as an individual object, known in OpenGIS terminology as a Feature. The FeatureType object, which can be requested from the Feature, represents the Feature's database scheme, containing detailed information about the properties of the Feature. Once the property definitions are available, the values of the properties can be requested by appropriate method calls. A specific method get_geometry() returns the Geometry of the Feature, as defined in the Geometry Object Model.

The Geometry Object Model contains specifications for 0-dimensional (Point), 1-dimensional (Curve) and 2-dimensional (Surface) geometry. All this Geometry is linear and planar. The Geometry Object Model also contains specifications for a group of geometries and specifies a general concept referred to as Geometry Collection, which can consist of heterogeneous geometry components.

Many GIS vendors have announced plans to develop an OGC SFS-compliant access interface for their GIS databases. An early demonstration of SFS-based interaction among four important GIS products was already given before the specification was officially published. The Distributed Spatial Technology Laboratory (DSTL) at BBN Technologies has been closely involved in the research and development resulting in the demonstration (DSTL 1998). The OpenGIS interface definitions have also been proposed as a reference model for a new Object Profile of the Spatial Data Transfer Standard (SDTS), a national standard in the USA (Arctur *et al.* 1998; pp. 403–425). The OGC has recently initiated a process to develop specifications for a GI-related Catalog Service and for interfaces to deal with raster data (Coverage). Together these specifications form a sound basis for a Web-based system architecture, as shown in figure 1.

## 3. NetGIS—a Java-based spatial Web browser

In this section we give a brief introduction to the technical features and functionality of NetGIS developed during this research (FGI 1998). The server side of the NetGIS application is currently based on a Java program simulating the behaviour of an SFS-compliant CORBA server—currently there is no real DBMS involved. In this article, however, we focus on the client-side environment.

### 3.1. *General*

In the following, a short presentation is given about how the Web technologies described above are utilized in our research. The aim of the project was to develop a Java applet for browsing GI in object form—most of the existing Web GIS solutions deliver spatial data as raster images. One of the benefits of having the GI delivered in object form is that a spatial object may then contain different types of data: coordinates as floating-point values, attributes in various data types, even images or other multimedia, thus enabling flexible use of varying visualisation methods in the browser. Figure 2 shows the appearance of the NetGIS applet, as viewed in the Netscape browser.

### 3.2. *Test data*

Visualisation of GI in NetGIS is divided into two main stages. In the first, the overall view of the selected territory of interest is visualised by using downloaded raster datasets as a background map (figure 2). In the second, object-based vector information can be requested via the OGC SFS interface from a remote server, the spatial extent of the query being indicated by the user (figure 3). Raster maps are

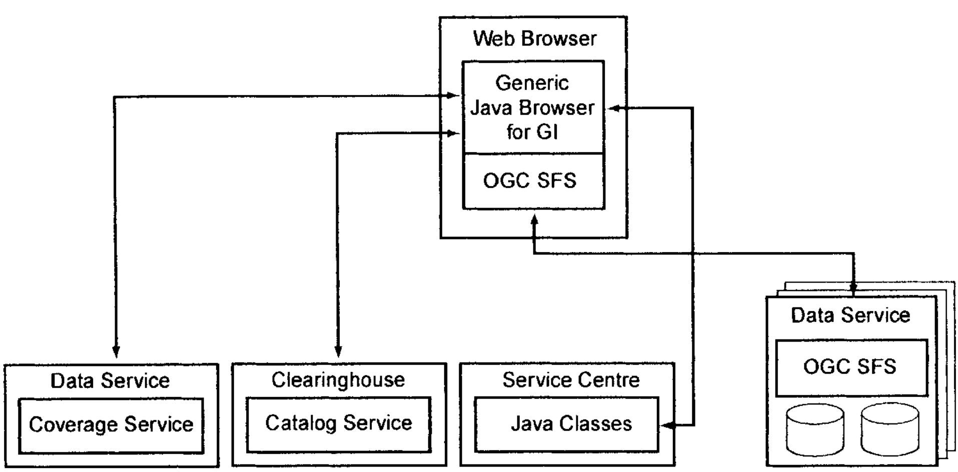


Figure 1. An OpenGIS Simple Features Specification-based system architecture on the Web.

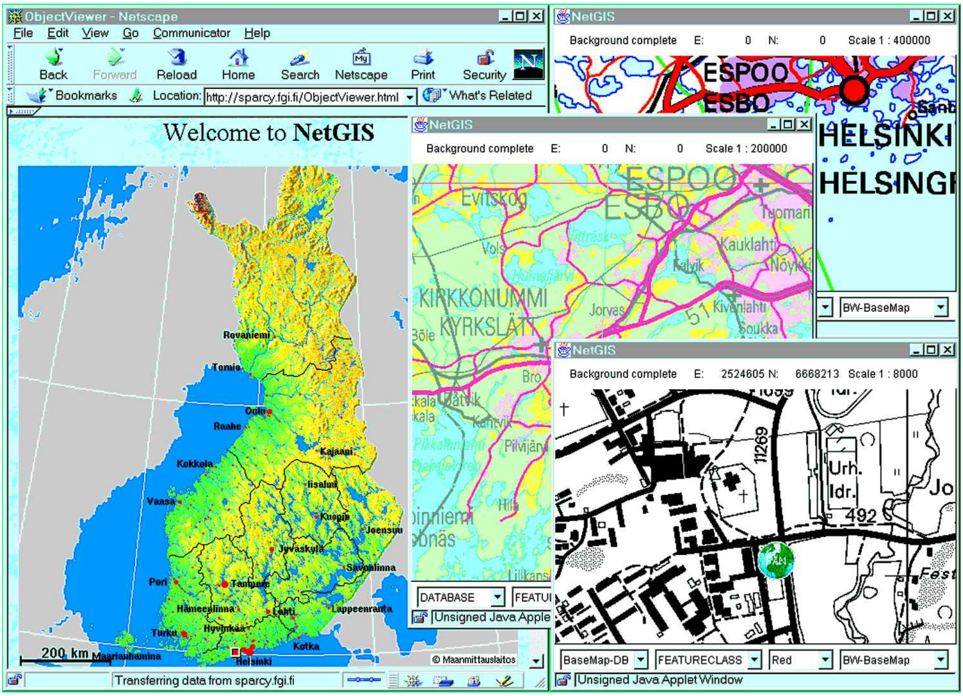


Figure 2. The visualization of geographical information in NetGIS using downloaded raster maps.

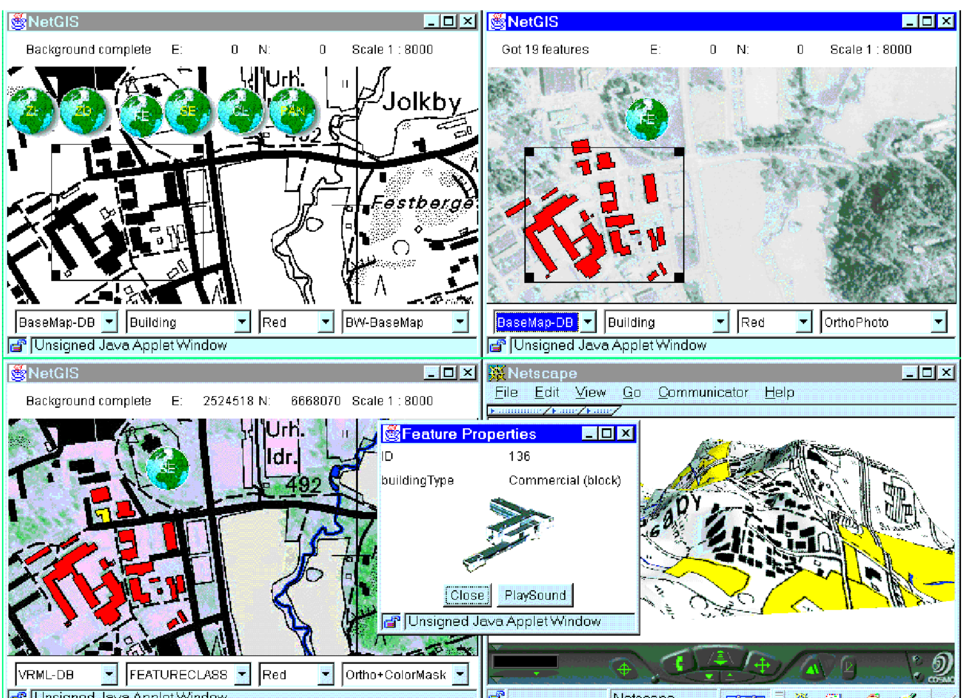


Figure 3. The visualization of geographical information as OpenGIS Simple Features.

produced from PerusCD, which includes a raster version of the traditional printed Base Map at a scale of 1: 20 000. Overview maps at scales of 1: 500 000, 1: 1 million and 1: 8 million from PerusCD are also utilised. Furthermore, digital orthophotos from OrthoCD have been used. PerusCD and OrthoCD are both products of the National Land Survey (NLS) of Finland (NLS 1998). A simple Java-based CORBA server was developed to simulate the behaviour of an OGC SFS-compliant data service. The vector data delivered by the server objects consist of building, road network and parcel information from the NLS.

### 3.3. *Graphical User Interface*

The Graphical User Interface of the map frame, popped up when the user selects an area by clicking the mouse on the image embedded in the HTML page, consists of three main parts (lower right in figure 2). The upper part of the frame is an information panel containing a message field, a coordinate display showing the current position of the cursor and the scale display. A map canvas where all map data are displayed in raster or vector format is located in the middle. The spatial extent of the map frame is also shown on the overview map; the user thus easily notices the national or global context and avoids large location errors when viewing maps. The user can interact with the map canvas by using special floating buttons (upper left in figure 3), which are displayed on top of the map. Once the user moves the cursor over the active button all the buttons will be displayed for selection. After selection only the active button is displayed, indicating the new functionality in use.

A data selection panel is located at the bottom of the frame. The fields of the panel allow the user to select a database from a list of available SFS-compliant services. Once selected, the service is requested for the available Feature classes and the corresponding list in the data selection panel updated accordingly. A Feature class can then be selected, the spatial extent of the query indicated on the map and the request sent to the server. The bottom panel also contains a selection list to define the colour used for rendering the Feature geometries and to select the type of raster dataset used as a background image.

### 3.4. *Raster data service*

A session begins when the user, browsing with a favourite Java-enabled Web browser, arrives at a Web page containing an embedded NetGIS applet tag. The applet is downloaded and started up in the user's browser. After waiting a moment while all required Java byte code is downloaded and interpreted at run time by the browser's JVM, the user will see an overview map (figure 2) that covers the complete area of Finland (1: 8 million overview map from PerusCD).

The user can then open the map frame by dragging a rectangle over the area of interest or by just clicking in the middle of the area. The client then sends an RMI request, containing the spatial extent of the defined area, to a specialised map sheet server. The server returns an appropriate patch of map index objects, each containing the spatial extent of the sheet and the URL for the map image. On the basis of this information, the applet can request the map images needed from an HTTP server. Before displaying maps on the screen, the images are assembled together. This enables flexible local panning and zooming of the raster map as a continuous background image. The raster dataset used is dependent upon the scale of the map display; thus, the user can zoom in through smaller-scale maps to more detailed ones. At the most detailed level, the Base Map at a scale of 1: 20 000 is available, as well as digital orthophotos, either as black-and-white images or combined with colour masks from the Base Map. The applet also handles the transformation from one projection zone to another in the Finnish national coordinate system.

### 3.5. *Object query*

Using the raster map as an orientation basis, the user can request geographical objects (Features in OGC terminology) from available databases (e.g. BaseMap-DB in figure 3). Communication between the client applet and a database server is carried out according to the OGC SFS. Once Feature Geometry is available to the client, the applet displays it on the top of the continuous raster map (e.g. buildings shown in red in figure 3). The geometry display colour can be defined and the geometries from different databases can thus be separated. The user is then given an opportunity to interactively choose an individual Feature to obtain detailed information on its properties (e.g. yellow building in figure 3). The Feature scheme is requested from the server using the SFS-defined method calls. The properties of the Feature can then be queried and displayed in a dynamically created menu. If the Feature contains a URL link to additional information, such as an image or an audio clip, the file can be requested and displayed (e.g. Feature Properties window in figure 3).

### 3.6. *Multimedia*

One of the goals of the present research has been to develop methods for visualising multimedia representations of geographical objects in the interactive Web environment. For example, buildings of cultural value may include images of their facades, or a central square may include voice representation to describe the traffic load at different times of the day. Voice and image representations are both supported as Java implementations in the current version of the NetGIS application.

Data in preprocessed Virtual Reality Modeling Language (VRML) file format can also be selected from a database list, and the selected area can be displayed in a separate VRML browser window (bottom right in figure 3). Currently, the possibility of visualising the GI objects or terrain in 3D models is restricted only to a limited VRML example. This example serves to simulate the future possibilities for visualisation. VRML and the new Java 3D™ programming interface, which is under development by JavaSoft, will offer an opportunity for 3D visualisation of GI. The new Java interface will deliver a comprehensive interactive 3D environment on a variety of platforms. The Java 3D and VRML Working Group (VRML Consortium 1998) has started trying to bring VRML and the Java 3D programming interface closer together and to improve their mutual interoperability.

## 4. **Discussion**

The NetGIS Java applet, together with the necessary server components, was developed at the Finnish Geodetic Institute as a proof-of-concept prototype application to investigate and demonstrate the viability of the described architecture for transmission and visualisation of geographical objects on the Web platform. The experience gained thus far strengthens the assumption of Java's suitability for implementation of a Web GIS application. Compared with existing raster-based solutions, delivery of GI as individual objects has several benefits. These include the opportunity to interact with a local software object, for instance, to request additional object-wise information. The object-based approach also enables sophisticated GIS analyses to be run locally on the client platform. The current situation in the Internet does not quite allow for the global deployment of NetGIS-like services—the most important obstacles being the need for a relatively fast Internet connection, and the still existing incompatibilities between different Java Virtual Machine implementations. In an intranet environment the architecture was proven to be realisable.

Several recent developments indicate the introduction of CORBA as a standard for distributed computing on the Web. The integration of Java and CORBA technologies allows communication between a Java client application and CORBA-compatible server applications via the CORBA object bus. The platform-independent nature of the Java code facilitates the distribution of active, locally executed client applications in the heterogenous Web environment. The CORBA standard enables access from the client to legacy systems, typically programmed in other languages.

A generic interactive browsing client, such as NetGIS, can be developed by establishing communication between the Java client applet and the server according to the OGC's SFS. These open standards—Java, CORBA and the OGC's SFS—enable the client to communicate with several information services via a standardized interface and create an integrated display of the spatial data received. The discussed architecture thus provides a platform for user-driven browsing and visualisation of object-based GI on the Web.

**Acknowledgments**

This research has been supported by the Finnish Ministry of Agriculture and Forestry under grant number 5995/416/96 for the Finnish Geodetic Institute. The geographical data and digital orthophotos were provided by the National Land Survey of Finland, which is much appreciated.

**References**

Arctur, D., Hair, D., Timson, G., Martin, E. P., and Fegeas, R., 1998, Issues and prospects for the next generation of the spatial data transfer standard (SDTS). *International Journal of Geographical Information Science*, **4**, 403–425.

Arnold, K., and Gosling, J., 1996, *The Java programming language* (Addison-Wesley).

DSTL (Distributed Spatial Technology Laboratory), 1998, Distributed spatial technology laboratory home page, http://javamap.bbn.com/

FGI (Finnish Geodetic Institute), 1998, Java-based spatial Web browser (NetGIS). Document on the Internet, http://www.fgi.fi/osastot/karto/projektit/netgis/vis_98/ngsh.html

JAVASOFT, 1998, JavaSoft home page, http://www.javasoft.com

Lehto, L., 1998, Java/CORBA integration—a new opportunity for distributed GIS on the Web. In *Proceedings of the ACSM Conference* (Baltimore: ACSM), pp. 474–481.

Lehto, L., Kähkönen, J., and Kilpeläinen, T., 1997, WWW-technology as means of transfer and visualization of geographic objects. In *Proceedings of the 18th ICA/ACI International Cartographic Conference* (Stockholm: ICA/ACI), pp. 681–689.

McKee, L., 1996, Digital maps: An essential part of every citizen's interface to the NII. OCS's position paper for the National Research Council Computer Science and Telecommunication Board's 'Toward an every-citizen interface to the NII' workshop. Document on the Internet, http://www.opengis.org/papers/citizen.html

NLS (National Land Survey of Finland), 1998, Finland on PerusCD. Document on the Internet, http://www.nls.fi/kartta/maps/peruscd.html.

OGC (Open GIS Consortium), 1997, OpenGIS® Specification marks new era of geoinformation access. Document on the Internet, http://www.opengis.org/info/press/pr_simple_features.htm

OGC (Open GIS Consortium), 1998, Open GIS Consortium home page, http://www.opengis.org

OMG (Object Management Group), 1997, What is CORBA? Document on the Internet, http://www.omg.org/corba/whatiscorba.html

VRML Consortium, 1998, The Java 3D™ and VRML working group. Document on the Internet, http://www.web3d.org

Vckovski, A., 1998, *Interoperable and distributed processing in GIS* (London: Taylor & Francis).

W3C (World Wide Web Consortium), 1998, HTML 4.0 Specification. W3C Recommendation, revised on 24 April 1998. Document on the Internet, http://www.w3.org/TR/REC-html40/

Zhang, L., and Lin, H., 1996, A client/server approach to 3D modeling support system for coast change study. In *Proceedings of the GIS/LIS Conference* (Denver, CO: ASPRS/ACSM/AAG/URISA/AM-FM), pp. 1265–1274.