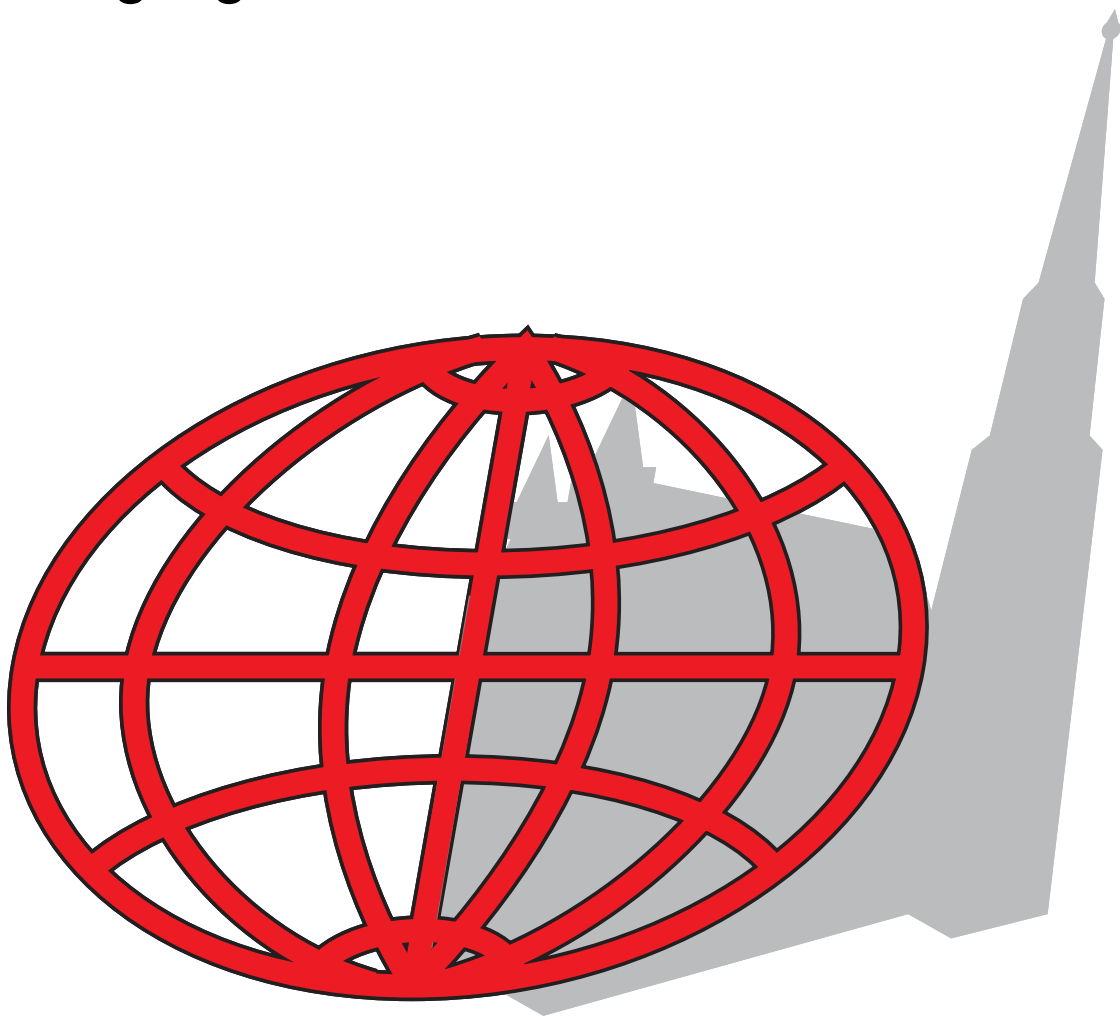




Geographic Information Science (GIS)

Wolfgang Kainz



© Wolfgang Kainz
Kartographie und Geoinformation
Institut für Geographie und Regionalforschung
Universität Wien
Universitätsstraße 7, A-1010 Wien
Tel. +43 (1) 4277 48640, Fax +43 (1) 4277 9486
wolfgang.kainz@univie.ac.at
<http://www.univie.ac.at/geographie>
Version 2.0 (25 February 2004)

Contents

0. Preface	ix
1. Geographic Information Systems	11
1.1 Geoinformatics and GIS	12
1.1.1 Terminology	13
1.1.2 Spatial Information Theory and the Geoinformatics Context	14
1.2 History of GIS	15
1.2.1 The Pioneers	15
1.2.2 The Breakthrough	16
1.2.3 The Ubiquitous GIS and the Road Ahead	17
1.3 Characteristics of Spatial Data	17
1.4 Functional Components of a GIS	18
2. Spatial Data	21
2.1 Metadata	22
2.2 Data Quality	23
2.3 Sources of Error	24
3. Data Models and Databases	25
3.1 Introduction to Information Processing	26
3.1.1 Real World Phenomena and Their Abstractions	26
3.1.2 Spatial Data and Information	27
3.2 Concepts of Space and Time	27
3.2.1 Pre-Newtonian Concepts of Space and Time	28
3.2.2 Classical Concepts of Space and Time	29
3.2.3 Contemporary Concepts of Space and Time	30
3.2.4 Concepts of Space and Time in Spatial Information Systems	30
3.3 The Real World and Its Models	31
3.3.1 Maps	31
3.3.2 Databases	32
3.3.3 Space and Time in Real World Models	33
3.4 Real World Models and Their Representation	34
3.4.1 Database Design	36
3.4.2 Topology and Spatial Relationships	37
3.4.3 Spatial Data Models	40
3.4.3.1 Field-based Models	40
3.4.3.2 Object-based Models	42
3.4.4 Spatiotemporal Data Models	42
3.4.4.1 Space-Time Cube Model	43
3.4.4.2 Snapshot Model	44
3.4.4.3 Space-Time Composite Model	44
3.4.4.4 Event-Based Model	44
3.4.4.5 Spatiotemporal Object Model	44
3.5 Summary	44
4. Spatial Data Structures	47
4.1 Point Data	48
4.1.1 Cell Method	48

4.1.2 Point Quadtrees	49
4.1.3 K-D Trees.....	50
4.1.4 PR Quadtrees	51
4.1.5 Grid Files	51
4.2 Line Data.....	52
4.2.1 Arc/Node Structure.....	52
4.2.2 Chain Codes.....	53
4.3 Area Data	53
4.3.1 Boundary Model.....	54
4.3.2 Region Quadtrees	54
4.4 Volume Data	55
4.5 Classification of Spatial Data Structures	55
5. Spatial Analysis	57
5.1 Data Organization and Spatial Queries	58
5.2 Maintenance and Analysis of the Spatial Data.....	58
5.3 Maintenance and Analysis of the Attribute Data	59
5.4 Integrated Analysis of Spatial and Attribute Data	59
5.4.1 Retrieval, classification, and measurement	59
5.4.2 Overlay Operations	59
5.4.3 Neighborhood Operations.....	59
5.4.4 Connectivity Operations	60
6. Technical Infrastructure	61
6.1 Introduction.....	62
6.2 Basic Hardware And Software.....	62
6.2.1 Computer Hardware	62
6.2.2 Peripheral Devices.....	63
6.2.3 Software	66
6.2.4 Computer Networks	67
6.2.5 Trends	68
6.3 System architecture	68
6.4 Computer networks.....	70
6.4.1 Transmission media	70
6.4.1.1 Twisted pair.....	70
6.4.1.2 Coaxial cable	71
6.4.1.3 Fiber optics	71
6.4.2 Wireless transmission.....	72
6.4.3 Network hardware	73
6.4.3.1 Local area networks (LANs).....	73
6.4.3.2 Metropolitan area networks (MANs)	74
6.4.3.3 Wide area networks (WANs).....	74
6.4.3.4 Internetworks	74
6.4.4 Network software	75
6.4.4.1 Layer Services	76
6.4.4.2 The OSI Reference Model	76
6.4.4.3 The TCP/IP Reference Model	78
6.4.4.4 Comparison between the OSI and TCP/IP Reference Models	80
6.5 The World Wide Web	80
7. Organizational Infrastructure	83
7.1 Database Technology	84
7.1.1 Database architectures	85

7.1.2 Data models.....	85
7.1.3 Distributed databases.....	86
7.2 Clearinghouse.....	86
7.3 Spatial Data Transfer.....	88
7.3.1 Spatial Data Transfer Process.....	88
7.3.2 Examples of Spatial Data Transfer Standards.....	89
8. GIS Technology in Organizations.....	91
8.1 Prerequisites and Problems.....	92
8.2 Implementation Strategy.....	92
9. References and Bibliography.....	95
10. Index.....	97

List of Figures

Figure 1: Geoinformatics context	14
Figure 2: Functional modules of a GIS.....	18
Figure 3: Platonic solids as building blocks of matter.....	29
Figure 4: Spatial modeling is a structure preserving mapping from the real world to a spatial model.....	34
Figure 5: Data modeling from the real world to a database, and from there to digital cartographic models and analogue products for visualization.	35
Figure 6: Rubber sheet transformation.....	37
Figure 7: Simplexes and simplicial complex. Features are approximated by a set of points, line segments, triangles, and tetrahedrons.	38
Figure 8: Cells and a cell complex. Spatial features are represented as topological images of cells.	39
Figure 9: Spatial relationships between two regions derived from the topological invariants of boundary and interior.	40
Figure 10: Two data layers in a field-based model.	41
Figure 11: Layers in an object based model.....	42
Figure 12: Grid representation for data in Table 7 (after SAMET 1990a).....	49
Figure 13: Point Quadtree obtained from insertion indicated in the text.....	49
Figure 14: A k-d tree for the data in Table 7 (after SAMET 1990a).....	50
Figure 15: A PR quadtree of the data points in Table 7 (after SAMET 1990a).....	51
Figure 16: Grid file partition and linear scales for the data of Table 7 (after SAMET 1990a).....	52
Figure 17: An arc/node structure	53
Figure 18: Chain code and sample line	53
Figure 19: Sample polygons and their topological data model. The outside region (background or world polygon) is denoted as O	54
Figure 20: Sample (a) region, (b) its binary array, (c) its maximal blocks, and (d) the corresponding quadtree (after SAMET 1990a)	54
Figure 21: Locational codes corresponding to the quadtree in Figure 20 (after SAMET 1990b).....	55
Figure 22: General computer architecture	63
Figure 23: Client/Server concept in computer networks	67
Figure 24: Stand-alone computers without cooperation.....	69
Figure 25. Client-server architecture	70
Figure 26: Coaxial cable	71
Figure 27: Fiber cable	71
Figure 28: Network topologies.....	73
Figure 29: Wide area network	74
Figure 30: Layers, protocols and interfaces.....	75
Figure 31: ISO/OSI reference model.....	76
Figure 32: ISO/OSI and TCP/IP layers	78
Figure 33: Database architecture.....	84
Figure 34: Models in database design	85
Figure 35: Top-down distributed database design	86
Figure 36: Federated database design	87

Figure 37: Clearinghouse concept	87
Figure 38: No standard versus a transfer standard.....	88
Figure 39 Spatial data transfer process.....	89

List of Tables

Table 1: Disciplines involved in spatial data handling	12
Table 2: Characteristics of spatial data.....	17
Table 3: Data input	18
Table 4: Data output and visualization.....	19
Table 5: Processing steps and sources of error (after ARONOFF, 1989)	24
Table 6: Data models and schemas in database design (the ANSI/SPARC architecture).....	36
Table 7: Sample point dataset (after SAMET 1990a).....	48
Table 8: Classification of data structures.....	56
Table 9: Types of queries	58
Table 10: Peripheral input and output devices overview	66
Table 11: Examples of spatial data transfer standards	89

0

Preface

*“Though this be madness,
Yet there’s method in it.”*

The Binary Bible of Saint Silicon

The first textbook on GIS was published in 1986. Today, we can find dozens of books dealing with GIS. Yet, it is difficult to select the right book that satisfies all needs. This lecture notes are the author’s view on what the basic scientific principles of GIS are that everybody should know who wants to use and understand today’s geographic information system software. They are the result of more than 20 years involvement in GIS and spatial data handling.

It is the intention to teach the principles of GIS independent of any particular software system. Only in this way, the student can abstract from the peculiarities of a particular system and open his/her mind to the fundamental principles of geoinformatics.

Any theory needs to be demonstrated with practical examples. This is also true for geographic information science. This lecture notes are designed in such a way that they can be used together with any reasonable GIS software to practice the principles learnt.

The notes start with a general introduction to the geoinformatics context and the design and structure of a GIS. The following chapter deals with metadata, data quality and sources of error in spatial data.

Chapter 3 introduces spatial data models and databases, which are followed by chapter 4 on spatial data structures. Experience shows that particularly with data models and data structures we often find a lot of confusion of terms and sloppy terminology. Therefore, much emphasis has been put to provide the student with a clear and clean treatment of these crucial concepts. Part of the material presented in chapters 4 and 5 come from an earlier lecture note on spatial data structures and algorithms that I have provided for ITC students. They were later revised and expanded by Rolf A. de By.

Chapter 5 describes in major lines the spatial analysis functions that we can expect in today’s GIS software. Issues on spatial data transfer and the introduction of GIS technology in organizations are presented in

chapters 6 and 7. This text is accompanied by slides that cover more and advanced subjects.

WOLFGANG KAINZ

Vienna, February 2004

Geographic Information Systems

The handling of spatial data usually involves processes of data acquisition, storage and maintenance, analysis and output. For many years, this has been done using analogue data sources, manual processing and the production of paper maps. The introduction of modern technologies has led to an increased use of computers and information technology in all aspects of spatial data handling. The software technology used in this domain is geographic information systems (GIS).

A general motivation for the use of GIS can be illustrated with the following example. For a planning task usually different maps and other data sources are needed. Assuming a conventional analogue procedure we would have to collect all the maps and documents needed before we can start the analysis. The first problem we encounter is that the maps and data have to be collected from different sources at different locations (e.g., mapping agency, geological survey, soil survey, forest survey, census bureau, etc.), and that they are in different scales and projections. In order to combine data from maps they have to be converted into working documents of the same scale and projection. This has to be done manually, and it requires much time and money.

With the help of a GIS, the maps can be stored in digital form in a database in world coordinates (meters or feet). This makes scale transformations unnecessary, and the conversion between map projections can be done easily with the software. The spatial analysis functions of the GIS are then applied to perform the planning tasks. This can speed up the process and allows for easy modifications to the analysis approach.

This chapter illustrates the geoinformatics context, reviews the historic development of GIS, explains the characteristics of spatial data, and describes the functional components of a geographic information system. The technology of geographic information systems was first developed in Canada in the 1960's. In only 25 years, it became a major business from large-scale applications down to the local level.

1.1 Geoinformatics and GIS

Spatial data handling involves many disciplines. We can distinguish disciplines that develop spatial concepts, provide means for capturing and processing of spatial data, provide a formal and theoretical foundation, are application-oriented, and support spatial data handling in legal and management aspects. Table 1 shows a classification of some of these disciplines. They are grouped according to how they deal with spatial information. The list is not meant to be exhaustive.

Table 1: Disciplines involved in spatial data handling

Characteristics of disciplines	Sample disciplines
Development of spatial concepts	Geography Cognitive Science Linguistics Psychology
Means for capturing and processing spatial data	Remote Sensing Surveying Engineering Cartography Photogrammetry
Formal and theoretical foundation	Computer Science Expert Systems Mathematics Statistics
Applications	Archaeology Architecture Forestry Geo-Sciences Regional and Urban Planning Surveying
Support	Legal Sciences Economy

The discipline that deals with all aspects of spatial data handling is called *geoinformatics*. It is defined as:

Geoinformatics is the integration of different disciplines dealing with spatial information.

Geoinformatics has also been described as “the science and technology dealing with the structure and character of spatial information, its capture, its classification and qualification, its storage, processing, portrayal and dissemination, including the infrastructure necessary to secure optimal use of this information” (GROOT, 1989). EHLERS and AMER (1991) define it as “the art, science or technology dealing with the acquisition, storage, processing production, presentation and dissemination of geoinformation.”

A related term that is sometimes used synonymously with geoinformatics is *geomatics*. It was originally introduced in Canada, and became very popular in French speaking countries. LAURINI and THOMPSON (1992) describe it as “the fusion of ideas from geo-sciences and informatics.” The term geomatics, however, was never fully accepted in the United States where the *term geographic information science* is preferred. GOODCHILD (1992) defines GIS research as “research on the generic issues that surround the use of GIS technology, impede its successful implementation, or emerge from an understanding of its potential capabilities.”

1.1.1 Terminology

Frequently used technical terms in spatial data handling related to GIS are:

geographic (or geographical) information system (GIS),
geo-information system,
spatial information system (SIS),
land information system (LIS), and
multi-purpose cadastre.

Geographic information systems are used by various disciplines as tools for spatial data handling in a geoinformatics environment.

There are many definitions for a geographic information system. The main characteristics, however, are the analytical functions that provide means for deriving new information from existing data. A GIS is defined as follows (ARONOFF, 1989):

A GIS is a computer-based system that provides the following four sets of capabilities to handle geo-referenced data:

1. input,
2. data management (data storage and retrieval),
3. manipulation and analysis, and
4. output.

Depending on the interest of a particular application, a GIS can be considered to be a data store (application of a spatial database), a tool-(box), a technology, an information source or a science (spatial information science).

Like in any other discipline, the use of tools for problem solving is one thing; to produce these tools is something different. Not all are equally well suited for a particular application. Tools can be improved and perfected to better serve a particular need or application. The discipline that provides the background for the production of the tools in spatial data handling is *spatial information theory* (or SIT) or geographic information science.

1.1.2 Spatial Information Theory and the Geoinformatics Context

Geographic information technology is used to manipulate objects in geographic space, and to acquire knowledge from spatial facts. Spatial information theory provides a basis for GIS by bringing together fields that deal with spatial reasoning, the representation of space, and human understanding of space:

Spatial reasoning addresses the inference of spatial information from spatial facts. It deals with the framework and models for space and time, and the relationships that can be identified between objects in a spatiotemporal model of real world phenomena.

Scientific methods for the *representation of space* are important for the development of data models and data structures to represent objects in spatial databases. Spatial databases are distinguished from standard databases by their capability to store and manage data with an extent in space and time (spatial data types).

The *human understanding of space*, influenced by language and culture, plays an important role in how people design and use GIS.

Figure 1 shows a general framework for geo-information. Here, spatial information theory is built on fundamental disciplines that provide the basis for the understanding of spatial phenomena, spatial cognition, concepts of space, as well as technical and formal tools, such as mathematics and computer science.

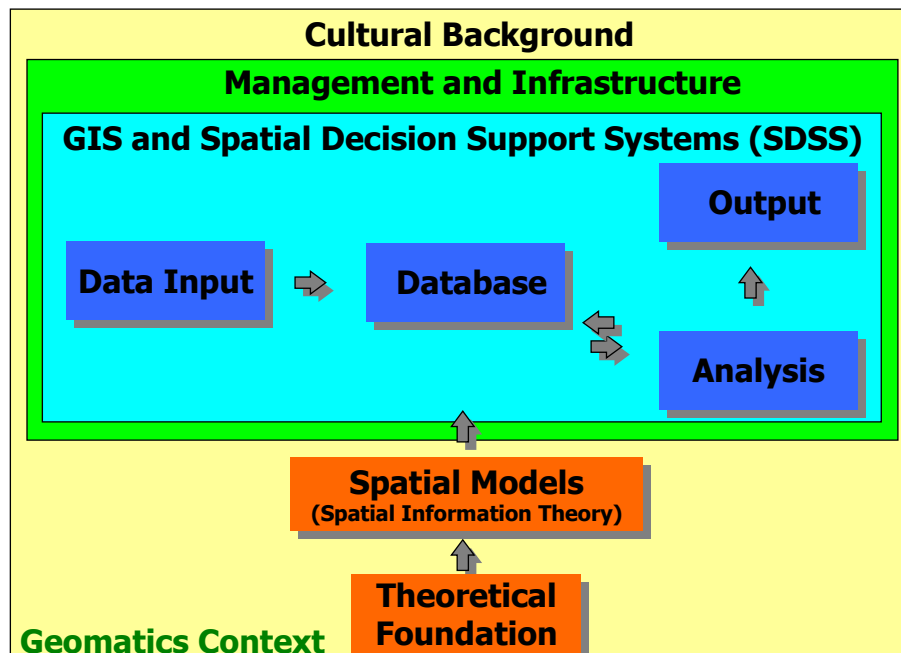


Figure 1: Geoinformatics context

The theory is used for the design of high-level models of spatial phenomena and processes. They are then mapped into conceptual,

logical and physical models for spatial databases. The database stands central in the geoinformatics environment. It is the database that holds the data; without it, no useful function can be performed. Data are entered into the database in input processes. Later, they are extracted from the database for spatial analysis and display.

The processes of data management, analysis and display are often supported by rules that are derived from domain experts. Systems that apply stored rules to arrive at conclusions are called rule-based or knowledge-based systems. Those that support decision making for space-related problems are known as *spatial decision support systems* (or SDSS). They are becoming increasingly popular in planning agencies and management of natural resources.

All these activities happen in a social, economic and legal context. It is generally referred to as spatial *information infrastructure*. Everything within this infrastructure and all concepts of space and time in turn are shaped and determined by the cultural background of the individuals and organizations involved.

1.2 History of GIS

When in the mid 1960s ROGER TOMLINSON developed the Canada Geographic Information System (CGIS), terms like quadtree, topology, object-orientation, workstation, personal computer, or geoinformatics were not known. For many of them this is true, because the objects that they would refer to, simply did not exist yet, or they did exist (as in the case of topology), but not in the context of *spatial* data handling. The efforts in building CGIS were driven by the need to overcome technical problems in getting maps into the computer. On the hardware side, a large-scale drum scanner had to be invented, and the well-known Morton order, a spatial indexing method, was developed.

1.2.1 The Pioneers

In the 1970s, developments mainly took place at universities in the United States, in Canada and the United Kingdom. The Laboratory for Computer Graphics and Spatial Analysis of the Harvard Graduate School of Design, or the Geographic Information Systems Laboratory at SUNY Buffalo achieved a worldwide recognition. Commercial agencies started to develop and offer GIS software. Among them were today's market leaders ESRI and Intergraph.

A growing awareness of the need for sound and stable structures to store and analyze map data became dominant in the late 1970s. This was the start of the successful series of Auto Carto conferences. The emphasis, however, was on *map data* rather than on spatial data. The idea of storing graphical features that are displayed on maps, in computer files was predominant. This is also reflected in the terminology used in the scientific literature of that time: map data model, map data structure or cartographic data structure are widespread. Abstract data types and spatial data models came later. They are children of the 1980s.

The search for stable and consistent representation structures of map data lead to the introduction of *topology* into GIS. Topology and the related graph theory proved to be effective and efficient tools to provide logically consistent 2-dimensional data representations. This success made many people believe that topology was invented by GIS researchers for GIS. In fact, topology is a branch of mathematics that

existed long before GIS was invented. The landmark paper by JAMES CORBETT on Topological Principles in Cartography became a milestone in research on spatial relationships (CORBETT, 1979).

1.2.2 The Breakthrough

The 1980s were shaped by the introduction and spread of the personal computer. For the first time in computing history, it was possible to have a computer on the desk that was able to execute programs that previously could only be run on mainframes. At the same time minicomputers, and later workstations, were widely available. Relational database technology became the standard.

Research on spatial data structures, indexing methods, and spatial databases made tremendous progress. This led to better and more reliable systems. Operational GIS software entered the market and was readily adopted mainly by government agencies for mapping, planning and design.

The first textbook on GIS appeared in 1986 (BURROUGH, 1986) and the International Journal of Geographical Information Systems was launched in 1987. The biennial series of Symposia on Spatial Data handling was started in 1984, and a similar one on Large Spatial Databases in 1989.

Despite all these promising developments, the need for a theoretical foundation of GIS became increasingly obvious. At the first International Symposium on Spatial Data Handling in Zurich, in 1984, JACK DANGERMOND, president of the Environmental Systems Research Institute, reported on the results of a workshop on GIS development that had been held one year before (DANGERMOND, 1984):

“There is at present no coherent mathematical theory of spatial relations. A small group of knowledgeable persons should be selected to define the problem thus created. Then a group of mathematicians should be selected to meet and refine the problem statement and to identify what means are best adapted to its solution. Research in this area should be funded and encouraged in such a way that significant progress might be made within about a ten-year period. Development and engineering implementations of the results of this research should be promptly supported.”

What was previously regarded as a tool for doing research (research *with* GIS), suddenly became a research topic itself (research *about* GIS). The research about GIS led to something that later would become known as Spatial Theory, Geographic Information Science, or Spatial Information Theory. Questions were asked like “How and why do GIS work?”, “How can we make them better?” or “What are the impediments to making them better?”

In 1988, the National Science Foundation of the United States approved the National Center for Geographic Information and Analysis (NCGIA), a consortium formed by the University of California Santa Barbara, the University of New York at Buffalo, and the University of Maine at Orono. Since its inception the NCGIA has started various research initiatives and provided major contributions to the progress in GIS research.

The European Science Foundation (ESF) initiated the GISDATA program that ran from 1993 to 1997. This was a European research program that resulted in a series of specialist meetings and book publications. In 1998 the Association of Geographic Information Laboratories in Europe (AGILE) was founded. Its mission is “to promote academic teaching and research on GIS at the European level and to ensure the continuation of the networking activities that have emerged as a result of the EGIS Conferences and the European Science Foundation GISDATA Scientific Programme¹.” AGILE organizes annual conferences that have been held so far at Enschede, The Netherlands, in 1998, Rome, Italy, in 1999, Helsinki, Finland, in 2000, Brno, Czech Republic, in 2001, Mallorca, Spain, in 2002, and Lyon, France, in 2003.

1.2.3 The Ubiquitous GIS and the Road Ahead

The 1990s can be characterized as a period of the breakthrough of object-orientation in system and database design, the recognition of geoinformatics as a professional activity, and spatial information theory as the theoretical basis for GIS. After a precursor under a different name in 1992, the first Conference on Spatial Information Theory (COSIT'93) was held in Elba, Italy, in 1993. This was the first in a biennial series of COSIT events. COSIT, the Symposia on Spatial Data Handling (SDH), and the Symposia on Large Spatial Databases² (SSD) are today's high-level forum for the presentation of research results in spatial information theory.

With the rise of the World Wide Web (WWW or the Web), new Internet protocols, such as the hypertext transfer protocol (HTTP), as well as easy to use interfaces (browsers), tools and languages (HTML, XML, Java), the network becomes the system. Internet applications and Internet GIS are the system of the future. The use of GIS is not any more restricted to mapping, planning and survey organizations. GIS enters medium and small businesses and new domains such as geo-marketing.

1.3 Characteristics of Spatial Data

A GIS deals with spatial data or *objects* (e.g., parcels, rivers, wells, ...), their *attributes* and characteristics (e.g., location, area, length, name, depth, ...) and the *relationships* between the objects (e.g., a parcel boundary follows a river, a well is located in a certain parcel, ...). The objects are stored in the database with *geometric primitives* (volumes, areas, lines, and points) and the relationships between them (*topology*). Spatial data have the following characteristics:

Table 2: Characteristics of spatial data

Spatial reference (geographic location, coordinates)	where?
Attributes (non-spatial)	what?
Spatial relationships (topology, metric, order)	in what relationship?
Temporal component (different concepts of time)	when?

¹ This mission statement is taken from the AGILE web site that can be found at <http://www.agile-online.org>.

² As of 2001 the conference is called Symposium on Spatial and Temporal Databases (SSTD).

1.4 Functional Components of a GIS

According to the definition a GIS always consists of modules for input, storage, analysis, display and output of spatial data. Figure 2 shows a diagram of these modules with arrows indicating the flow of data in the system. For a particular system, each of these modules may provide more or less functions. However, if one would be completely missing the system should not be called a geographic information system. The most important component of GIS software is the functional complex of functions for spatial analysis. If this functionality is poorly developed, we cannot call such a system a GIS.

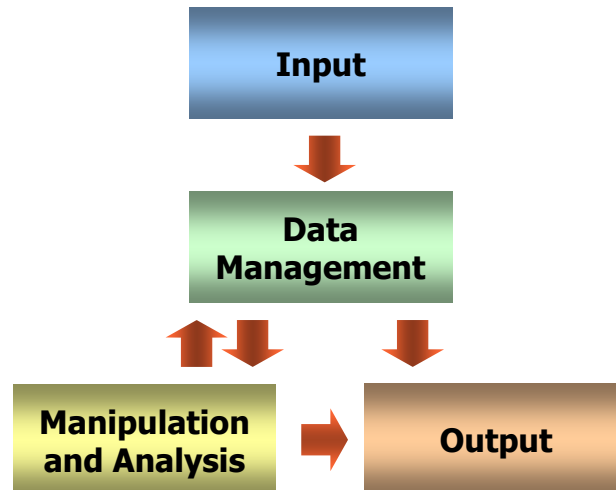


Figure 2: Functional modules of a GIS

The functions for data input are closely related to the disciplines of surveying engineering, photogrammetry, remote sensing, and the processes of digitizing, i.e., the conversion of analog data into a digital representation. Today, digital data on different media or on a computer network are used increasingly. Table 3 lists the methods and devices used in the data input process.

Table 3: Data input

Method	Devices
manual digitizing	coordinate entry via keyboard digitizing tablet with cursor or mouse (digital) photogrammetry
automatic digitizing	scanner
semi-automatic digitizing	line following devices
input of available digital data	magnetic tape CD ROM networks

Data output is closely related to the disciplines of cartography, printing and publishing. Table 4 lists different methods and devices used for the output of spatial data. Cartography and scientific visualization make use of these methods and devices to produce their products. The importance of digital products (datasets) is increasing and data dissemination on digital media or on computer networks becomes extremely important.

In both processes, data input and data output, the Internet, and Internet technology have a major share. The World Wide Web plays the

role of an easy to use interface to repositories of large datasets. Aspects of data dissemination, security, copyright, and pricing require special attention. Spatial information infrastructure deals with these issues.

Table 4: Data output and visualization

Method	Devices
Hardcopy	printer plotter (pen plotter, ink-jet printer, thermal transfer printer, electrostatic plotter) film writer
Softcopy	computer screen (CRT)
Output of digital datasets	magnetic tape CD ROM network

Spatial Data

Users of GIS need to know about certain characteristics of the data they are using for their applications. Many institutions provide a type of service that helps to select the right dataset for a particular application. Such institutions are called clearinghouse.

The need to have a common set of characteristics of data has led to standardization efforts in various organizations and at different levels. The International Organization for Standardization (ISO) has established a technical committee (ISO/TC 211) on geographic information/geomatics that deals (among others) with the standardization of metadata and data quality.

The standardization body of the European Union, CEN, is active in developing a series of standards concerning the same issues. On a national level, organizations like the Federal Geographic Data Committee (FGDC) of the United States of America have also developed standards related to metadata and other processes in spatial data handling. Currently, the ISO and CEN standards are draft standards or pre-standards, awaiting their final approval.

This chapter introduces and explains the concept of metadata, describes the relevance of data quality as one of the metadata elements, and explains various sources of error that can affect the fitness for use of a dataset.

2.1 Metadata

Users looking for datasets for a particular application need to know about certain characteristics of these data in order to make the right selection. This type of information is called metadata.

Metadata are data about the content, quality, condition, and other characteristics of data. Metadata are organized in *sections* containing one or more metadata *entities*, which in turn are a collection of similar metadata *elements*.

A *feature type* is a class of features with common characteristics (e.g., country). With *feature*, we mean a representation of a real world phenomenon (e.g., Austria). A *feature attribute type* is a characteristic relevant for features belonging to a certain feature type (e.g., area, population,...). A *feature attribute* is a characteristic of a specific feature. It has a name from its feature attribute type and a value from the value domain associated with that feature attribute type (e.g., area 83,858 km², population 8,2 million). A logical link between features is called a *feature relationship* (e.g., the neighboring countries of Austria are Germany, Czech Republic, Slovak Republic, Hungary, Slovenia, Italy, Liechtenstein, and Switzerland).

We can distinguish between eleven sections (packages) that contain the metadata elements. The following is taken from the ISO 19115:2003 *Geographic Information - Metadata* standard.

Identification Information. This helps to uniquely identify a dataset.

Constraint Information. This contains information concerning the restrictions placed on data.

Data Quality Information. Contains a general assessment of the quality of the dataset (see section 2.2 below).

Maintenance Information. Contains information about the scope and frequency of updating data.

Spatial Representation Information. Contains information concerning the mechanisms used to represent spatial information in the dataset.

Reference System Information. This is a description of the spatial and temporal reference used in the dataset.

Content Information. This section contains definitions and descriptions of the feature types together with their feature functions, attributes, and relationships occurring in the dataset.

Portrayal Catalogue Information. This section contains information identifying the portrayal catalogue used.

Distribution information. Contains information about the distributor of the dataset and about options for obtaining the dataset.

Metadata Extension Information. Contains information about the user-specified extensions.

Application Schema Information. Contains information about the application schema used to build a dataset.

2.2 Data Quality

It is very important to know about the quality of data used in a GIS application. Often this is expressed as fitness for use.

Data quality is the totality of characteristics of a product that bear on its ability to satisfy stated and implied needs (ISO 8402). Data quality is documented by the *data quality elements* described here, and tested through *data quality measures* that result in *data quality results*.

The following is taken from the ISO 19113:2002 *Geographic Information – Quality Principles*, and ISO 19114:2003 *Geographic Information – Quality Evaluation Procedures* standards.

The standard distinguishes between *data quality elements* (quantitative) and *data quality overview elements* (qualitative). The data quality elements are described as follows:

Completeness describes the presence and absence of features, their attributes and their relationships. Data quality measures are measures of difference between the specified items and the actual items in a dataset.

Logical consistency describes the degree to which data is stored in accordance with the structure of a dataset or the degree to which the correct encoding of feature attributes into a dataset are in accordance with a dataset's production specification. A data quality measure for logical consistency is the degree of consistency expressed, for instance, as the percentage or number of violations.

Positional accuracy describes the accuracy of the position of features in accordance with a dataset's product specification. An error statistic (standard deviation, % confidence) is used as a data quality measure.

Temporal accuracy describes the accuracy of the temporal features and feature attributes in accordance with a dataset's product specification. Data quality measures are error statistics (standard deviation, % confidence), percentage, or conformance.

Thematic accuracy describes the accuracy of the classifications of the features and feature attributes of a dataset in accordance with a dataset's product specification. Data quality measures common for thematic accuracy are error statistics (standard deviation, % confidence), conformance indicators, or a (misclassification) matrix.

Data quality overview elements that give a qualitative description of the quality of a dataset are:

Purpose describes the rationale for creating the dataset and contains information about its intended use.

Usage describes the application(s) for which the dataset has been used by producers or users.

Lineage describes the history of the dataset and the processes that were involved in its production (life cycle).

For both quantitative and qualitative quality elements, the standard allows to define additional user-defined quality elements.

2.3 Sources of Error

All geographic data contain some degree of error. During every processing step in spatial data handling there are many possibilities to influence the quality of data or products. Often these sources of error are not easy to detect, or they are hidden by deficiencies of instruments, software or personnel.

These errors are then propagated through repeated processing steps and can sometimes render a product unsuitable for a particular purpose that it was originally meant for. Table 5 lists the most important sources of error in the different processing steps.

Table 5: Processing steps and sources of error (after ARONOFF, 1989)

Processing Step	Error Source
data capture	errors with field data capture errors in maps that are used as data sources errors in remotely sensed data
data entry	inaccuracies introduced during digitizing by the operators or the hardware inaccuracies inherent in the objects (e.g., fuzzy boundaries digitized as clear cut lines)
data storage	insufficient numerical accuracy insufficient spatial accuracy
data manipulation	inappropriate class boundaries error propagation through computer arithmetic slivers introduced by overlay processes
data output	inaccuracy of output devices instability of output media
use of results	insufficient technical knowledge wrong use of results

Data Models and Databases

When we want to represent phenomena of the real world in a GIS we need to design a model that describes these phenomena in the best way for our particular purposes. This process is called spatial modeling. We store the data in a GIS database (spatial database).

In the design process of a database, we go through several modeling steps to produce database schemas from high level software and hardware independent conceptual and logical schemas to the implementation with particular database software.

This chapter introduces the concepts of real world phenomena and their representations in spatial databases. Concepts of space and time relevant for spatial data play an important role in spatial modeling. Several approaches to spatial data modeling and models of time used in GIS are discussed.

3.1 Introduction to Information Processing

We live in a constantly changing world. What we perceive with our senses are processes, states and events that happen or exist in the real world. They may be natural or man-made, and are called real world phenomena. In our brains, we process the input from our senses, which leads to mental models, learning, cognition, and knowledge. Everything that happens in the real world and that we process through our senses leads to models of reality that we create for ourselves. Usually, we agree (tacitly) on common principles and interpretations derived from experience, research or learning that enable us to reach a consensus (at least largely) on the perception of real world phenomena.

It is evident that for many phenomena such an agreement is obvious because of the same educational, cultural or societal context, or simply by common sense. For instance, there is hardly any dispute about the perception of a tree, water body or mountain, although we give them different names according to our languages. The essence, however, is well understood.

We will not discuss here the fundamental philosophical questions of “What can we know?” or “How real is reality?” They have been the subjects of philosophical contemplation since the dawn of humankind, and the answers largely depend on the philosophical or religious orientation of the persons involved. What we state here, however, is that in our minds we create models of the real world. The assumption (based on experience and common sense) is that within a given context the individual mental models coincide to such extent that we perceive the same reality.

3.1.1 Real World Phenomena and Their Abstractions

In the real world, we distinguish between natural and man-made phenomena. *Natural phenomena* exist independently from human actions and are subject only to the laws of nature. Examples are the landscape (topography), the weather, or natural processes that shape and influence them. *Man-made phenomena* are objects that have come into existence by human activities through construction or building processes.

Based on these phenomena we develop high-level abstract *models* for particular purposes and applications. Features (abstractions of phenomena) populate the models that are usually organized in *layers*. Examples of such models are a cadastre, topography, soil, hydrography, land cover, or land use.

The fact that these models are abstractions of the real world can be illustrated by the example of a cadastre. Let us assume that a cadastre is a legal and organizational framework for the handling of land. It is a very important, clearly described and understood concept. Yet, we do not see a cadastre when we look around us. What we see are real world phenomena such as buildings, roads, fences around pieces of land, and people. A cadastre abstracts from certain phenomena and their relationships to create something new that is real in a given context.

Layers are an ordering principle for real world phenomena. Again, when we look around us, we do not see the world in layers. Yet, we are used to organize phenomena of the real world in such a way that we

classify them according to a perceived purpose or characteristic into subsets (layers) that allow us to deal with them efficiently.

3.1.2 Spatial Data and Information

In order to conceptualize mental models of the real world, we need to categorize the phenomena we observe. These phenomena exist in space and time and have therefore a spatial (geometric) and a temporal extent. They possess certain thematic characteristics that allow us not only to refer to them in terms of spatiotemporal, but also according to thematic information. The thematic information of real world phenomena is the basis for the definition of layers.

Humans perceive signals through their senses, process them and extract information that leads to knowledge and wisdom. Here, we focus on spatial information, i.e., information concerning phenomena with a spatiotemporal extent. It is obvious that thematic information is an integral part of spatial phenomena.

Data are representations of information in a computer. Spatial data refer to spatial information that we store in a computer for processing and analysis.

The following example illustrates the principle. Assume that you stand on top of a hill and are looking at the landscape surrounding you. What you see are meadows, fields, trees, and roads. The meadows are green with grass, the fields are yellow, the trees are green, and the road is covered with brownish-black asphalt. Your brain has processed the optical signals that you receive through your eyes and your mind recognizes the phenomena observed. You also have assigned attributes to them such as color, (relative) size and you might remember that five years ago the road was not paved, and what is a field now has been a forest.

Since you want to build a land cover database of this area, you need to store a representation of the phenomena in a computer database. To achieve this you need to define features, collect (spatial and attribute) data about them and enter them into a database. Geographic information system (GIS) software will be used for processing and analysis.

3.2 Concepts of Space and Time

Space and time are two closely related concepts that have been the subject of philosophical and scientific consideration since the dawn of mankind. The space that humans live in is the three-dimensional (Euclidean) space as a frame of reference for our senses of touch and sight. Of all possible spaces, this is the only space that is illustrative and that we perceive as being real.

Time is a measure for change in our immediate experience. Usually we assume time to be of a continuous linear nature extending from the past, through the present into the future.

Space and time (at least as we perceive them) are so well known and appear to be given beyond any doubt that we hardly ever contemplate their structure and characteristics in our everyday lives. When we deal with information systems that process and manipulate spatiotemporal

features, we need a clear and well-understood model of space and time. The following sections describe how concepts of space and time developed in Western philosophy and physics. We discuss them according to three epochs: (i) pre-Newtonian concepts, (ii) the Newtonian and classical concepts, and (iii) contemporary concepts of space and time.

3.2.1 Pre-Newtonian Concepts of Space and Time

The concepts of space and time of this epoch are mainly dominated by the ideas of Greek philosophers about the logical conditions for things to change and the structure of the world in which change occurs.

HERACLITUS (around 500 BC) of Ephesos (western Turkey) studied the problem of change, i.e., how can the identity of things be preserved when they change. He stated that everything flows, nothing remains, and the only thing that really exists is *change (processes)*. “Everything flows” and “We cannot step into the same river twice” are attributed to him. *Identity of opposites*, linked by processes, is his postulate.

At about the same time, PARMENIDES of Elea (southern Italy) developed a completely opposite philosophy of the *non-existence of the void*. He postulates (through deductive reasoning) that *change does not exist* and that the real world (the real being) is plenum (a solid complete compact being), immutable, without change and eternal. A void (or empty space, i.e., something non-existent) does not exist. What we perceive as change is a delusion of our senses. PARMENIDES’ ideas were further developed and “proven” by his student ZENO.

One of ZENO’s famous “proofs” that change and movement cannot exist is known as the race between Achilles and the tortoise. The tortoise gets a head-start and begins the race at point B, whereas Achilles starts from point A. When Achilles reaches point B, the tortoise already has moved on to point C. When Achilles reaches point C, the tortoise has moved on to another point, and so on. The lead of the tortoise gets smaller and smaller, *ad infinitum*. We get an infinite number of (smaller and smaller) leads.

The argument is now: In order to reach the tortoise, Achilles must catch up an infinite number of (finite in length) leads. It is impossible to run this infinite number of short distances, because Achilles would have to run infinitely far (or forever). Therefore, it is impossible for Achilles to catch the tortoise. Since we can easily catch a tortoise when we would run against it, we end up with a paradox. This proves that the assumption that movement and change are real leads to contradictions. Therefore, movement and change are impossible.

DEMOCRITUS (460–370 BC) did not accept the non-existence of change as postulated by PARMENIDES. Space is an absolute and empty entity existing independently from the *atoms* that fill the space. Atoms are indivisible real things; they are immutable and eternal, and have different size and weight. There is no empty space within atoms. An atom is a plenum. Objects are formed as a collection of atoms. The importance of the Atomist theory is evident in today’s modern particle physics.

The Greek mathematics was strongly dominated by the Pythagorean number theoretic approach. It was essentially arithmetic based on (philosophical) properties of numbers, counting and the ratios between numbers. The discovery of irrational numbers such as $\sqrt{2}$ (the length of the diagonal in the unit square) shook the foundation of Greek

mathematics that was based on counting in natural units. The need for a truly geometrical description of the world became apparent.

The great philosopher PLATO (427–347 BC) and one member of his school, EUCLID (at 300 BC), laid the foundation for a new geometric modeling of the real world. This geometric model of matter is based on right triangles as atoms and solids built from these triangles. Matter consists of four elements: earth, air, fire, and water. Each element is made of particles, i.e., solids (see Figure 3) that in turn are made from triangles. Transformations between the elements fire, air and water are possible through geometric transformations. Earth cannot be transformed.

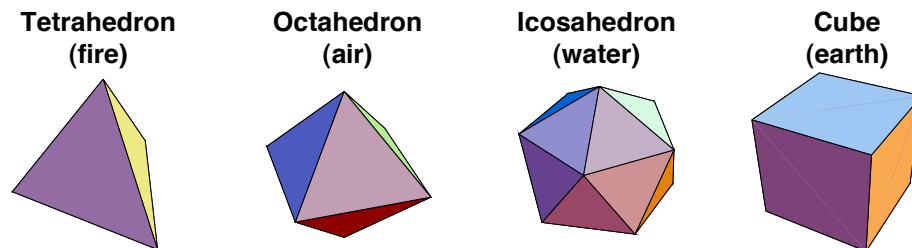


Figure 3: Platonic solids as building blocks of matter.

In his *Elements*, EUCLID developed a mathematical theory of geometry that remained valid until the late 19th century. Euclidean geometry was considered a true description of our physical world until it was discovered that many consistent geometric systems are possible, some of them non-Euclidean, and that geometry is not a description of the world but yet another formal mathematical system with no necessary reference to real world phenomena.

3.2.2 Classical Concepts of Space and Time

The time between the classical Greek period and the rise of modern science was dominated by the philosophy and teaching of ARISTOTLE (384–322 BC). According to his ideas, empty space is impossible, and time is the measure of motion with regard to what is earlier and later. Space is defined as the limit of the surrounding body towards what is surrounded. Space is a place of all things that occupy a place in space.

Following from this approach space can be conceptualized in two possible ways:

Absolute space. Space as a set of places. It is an absolute real entity, the container of all things. Its structure is fixed and invariable. Generally, this is considered the space as described by the Euclidean geometry.

Relative space. Space is a system of relations. It is the set of all material things, and relations are abstracted from them. Space is a property of things or things have spatial properties.

The rise of modern science took shape in the works of Nikolaus COPERNICUS (heliocentric system: the sun seen as the center of our planetary system), Johannes KEPLER (mathematical foundation of the heliocentric system), and Galileo GALILEI (foundations of mechanics) in the 16th and 17th centuries.

Isaac NEWTON (1643 – 1727 AD) was a brilliant scientist (dynamic theory) and philosopher. In his philosophy, he was an outspoken proponent of the concept of absolute space although it strictly

contradicts his dynamics theory. The concept of absolute space remained dominant until the late 19th century.

Gottfried Wilhelm LEIBNITZ (1646–1716) on the other hand sustained the concept of relative space. For him, space is a system of relationships between things. It is interesting to note that both NEWTON and LEIBNITZ are the founders of mathematical calculus.

One of the greatest philosophers, Immanuel KANT (1724–1804), claimed that space and time are not empirical physical objects or events. They are merely *a priori* true intuitions, not developed by experience, but used by us to relate and order observations of the real world. Space and time have *empirical reality* (they are absolute and a priori given) and *transcendental idealism* (they belong to our conceptions of things but are not part of the things). We cannot know anything about the things as such. In this regard, KANT can be regarded as a proponent of absolute space, yet in a far more elaborate and sophisticated way than the previous philosophical approaches.

3.2.3 Contemporary Concepts of Space and Time

The development of modern physics (field theories, theory of relativity, quantum theory) and mathematics (non-Euclidean geometries) lead to the conclusion that traditional Euclidean geometry (describing the three-dimensional space of our perception) is only an approximation to the real nature of the world.

The field theories (Michael FARADAY and James Clerk MAXWELL) lead to the assumption that space is not empty, but is filled with energy. Therefore, a material existence of space is strongly supported by these theories.

As a consequence of the special and general theory of relativity by Albert EINSTEIN (1879–1955), space and time cannot anymore be considered as two separate entities. We speak of space-time, which is considered a four-dimensional curved space that can only be described by non-Euclidean geometry. Quantum mechanics states the principle of uncertainty and the discrete character of matter and energy. It has become evident that the space of our perception is not necessarily identical with the microscopic (sub-atomic) space or the space of cosmic dimensions.

3.2.4 Concepts of Space and Time in Spatial Information Systems

Spatial information is always related to *geographic space*, i.e., large-scale space. This is the space beyond the human body, space that represents the surrounding geographic world. Within such space, we constantly move around, we navigate in it, and we conceptualize it in different ways. Geographic space is the space of topographic, cadastral, and other features of the geographic world. Geographic information system technology is used to manipulate objects in geographic space, and to acquire knowledge from spatial facts.

Geographic space is distinct from small-scale space, or tabletop space. In other words, objects that are smaller than us, objects that can be moved around on a tabletop, belong to small-scale space and are not subject of our interest.

The human understanding of space, influenced by language and cultural background, plays an important role in how we design and use tools for the processing of spatial data.

In the same way as spatial information is always related to geographic space, it relates to *geographic time*, the time whose effects we observe in the changing geographic world around us. We are less interested in pure philosophical or physical considerations about time or space-time, but more in the observable spatiotemporal effects that can be described, measured and stored in information systems.

3.3 The Real World and Its Models

As mentioned in the previous sections we always create models of the real world in our minds. When we want to acquire, store, analyze, visualize, and exchange information about the real world, we use other media and means than just interpersonal communication. We need representations of our mind models, i.e., models of the real world that can be used to acquire, store, analyze and transfer information about real world data.

The most common of these models are—in historic sequence—maps and databases. Both have distinct characteristics, advantages and disadvantages. Whereas maps usually have been used to picture real world phenomena, databases can be used to represent real and virtual worlds.

Real worlds are subsets of the reality that we perceive. Virtual worlds are computer generated ‘realities’ that exist only as potentialities with no counterpart in the real world. Yet, we can visualize them, navigate through them and perceive them as ‘real’ (therefore the term virtual reality). There is no difference between real and virtual world models with regard to their representation. The only difference is that the former is a model of something that exists in the real world and the latter is a model of something that exists only in a virtual (physically non-existent) world.

3.3.1 Maps

The best known (conventional) models of the real world are maps. Maps have been used for thousands of years to represent information about the real world. Today, they are usually drawn on paper or other permanent material and function as data storage and visualization medium. Their conception and design has developed into an art and science with a high degree of scientific sophistication and artistic craftsmanship. Maps have proven to be extremely useful models of reality for many applications in various domains.

Yet, maps are two-dimensional (flat) and static. It is not easy to visualize three-dimensional dynamic features without considerable abstractions in the spatial and temporal domain. We distinguish between topographic and thematic maps. Other cartographic products that are not maps are often used to represent three-dimensional and dynamic phenomena. Such products are, for instance, block diagrams, animations, and panoramic views.

A disadvantage of maps is that they are restricted to two-dimensional static representations, and that they always are displayed in a given scale. The map scale determines the spatial resolution of the graphic feature representation. The smaller the scale, the less detail a map can show. The accuracy of the primary data, on the other hand, puts limits to the scale in which a map sensibly can be drawn. The selection of a proper map scale is one of the first and most important steps in a map conception.

A map is always a graphic representation at a certain level of detail, which is determined by the scale. The process to derive less detailed representations from a detailed one is called *map generalization* (or *cartographic generalization*). Map sheets have physical boundaries, and features spanning two map sheets have to be cut into pieces.

Cartography as the science and art of map making functions as an interpreter translating real world phenomena (primary data) into correct, clear and understandable representations for our use. Maps also become a data source for other maps.

With the advent of computer systems, analogue cartography became digital cartography. It is important to note that whenever we speak about cartography today, we implicitly assume digital cartography. The use of computers in map making is an integral part of modern cartography. The role of the map changed accordingly. Increasingly, maps lose their role as data storage. This role is taken over by databases. What remains is the visualization function of maps.

When we look back into the history of digital cartography and geographic information systems, we see that originally maps were considered the main data source for GIS databases. To transfer the contents of a map into a computer database was the major goal. We observe this also in the scientific literature of that time. The terms *map data model* and *map data structure* were widely used. It was not clearly understood that we actually want to store representations of real world phenomena (primary data) and not map data (secondary data). In short, to store map data into a database instead of primary data means to create the model of a model.

3.3.2 Databases

Spatial databases store representations of spatial phenomena in the real world to be used in a geographic information system. They are also called GIS databases. In the design of a database, we distinguish between three different levels of definition. A language that we use to define the database is called a *data model*; each level typically has its own data model. The data model used at the level closest to the end-users—and relatively far from any computer implementation—is called a *conceptual data model*. In our context, it is used for *spatial data modeling*. The intention is to define which concepts of interest exist in the application domain for which a database is being designed, and what their relations are. Such a definition identifies the types of things relevant for a particular application, for example, a cadastral administration, a landslide hazard analysis system, a geologic hypermap toolbox. A commonly used conceptual data model is the entity-relationship (ER) model; it uses primitives like entity type to describe independently existing entities, relationship type to define relationships between entities, and attributes to describe characteristic values of entities and relationships. The complete database definition is called the (conceptual) *database schema*. It can be likened to a story, written in the language that is the data model. Other, more implementation-oriented, data models will be discussed in Section 3.4.1.

The assumption for the design of a spatial database schema is that spatial phenomena exist in a two- or three-dimensional Euclidean space. All phenomena have various relationships among each other and possess spatial (geometric), thematic and temporal attributes (they exist in space and time). Phenomena are classified into thematic layers depending on the purpose of the database. This is usually described by

a qualification of the database as, for example, cadastral, topographic, land use, or soil database.

The representations of spatial phenomena (spatial features) are stored in a scale less and seamless manner. *Scale less* means that all coordinates are world coordinates given in units that are normally used to reference features in the real world (geographic coordinates as latitude and longitude, or metric units in meters). From there, calculations can be easily performed and any (useful) scale can be chosen for visualization.

It must be noted, however, that scale plays a role when data are captured from maps as data source. Here, the scale of the source map determines the accuracy of the feature coordinates in the database. Likewise, the accuracy of measurements in field surveys determines the quality of the data. If the coordinates are given in units other than geographic coordinates, information concerning the spatial reference system should also be present.

A *seamless* database does not show map sheet boundaries or other partitions of the geographic space other than imposed by the spatial features themselves.

The advantages of databases as real world models are that they are scale-less, potentially three-dimensional, dynamic, and seamless. It is easy to query a database, and to combine data from different layers (spatial join or overlay). Spatiotemporal databases consider not only the spatial and thematic but also the temporal extent of the features they represent. Various spatial, temporal and spatiotemporal data models have been developed. They will be discussed in Section 3.4.4.

The attentive reader will have noted our threefold use of the word 'model'. This, perhaps, may be confusing. Except as a verb, where it means 'to describe' or 'to represent', it is also used as noun. A 'real world model' is a representation of a number of phenomena in the real world, usually to enable some type of administration, computation and/or simulation. It is the result of the activity of 'modeling'. A 'data model', on the other hand, is a database language used as a tool in database design, in an activity called 'data modeling'. The result of that activity is not a data model, but a database schema, an abstract definition of the contents of the (future) database. A database schema can be viewed as a special kind of real world model, but it is abstract because it identifies only types of things in that real world, and not the things themselves. In a cadastral setting, for instance, it would identify Parcel, Owner, Title as three entity types, without identifying any actual parcels, owners or titles. This would be done in the database itself. Therefore, we might say that a database schema is an occurrence-independent real world model.

3.3.3 Space and Time in Real World Models

In modern physics, it is common to speak of space-time to express the close connection that exists between space and time according to the special and general theory of relativity. Here, we do not consider the physical characteristics of space and time, but the (simplified) ways of representing spatial phenomena in a GIS database.

In general, *modeling* can be described as creating a structure preserving mapping (*morphism*) from a *domain* to a *co-domain*. In

our case, the domain is the real world, and the co-domain is a real world model. Such a mapping normally creates a ‘smaller’ (i.e., abstracted, generalized) image of the original. Structure preserving means that the elements of the co-domain (spatial features) behave in the same (however simplified or abstracted) way as the elements of the domain (spatial phenomena). Figure 4 illustrates the principle of *spatial modeling*.

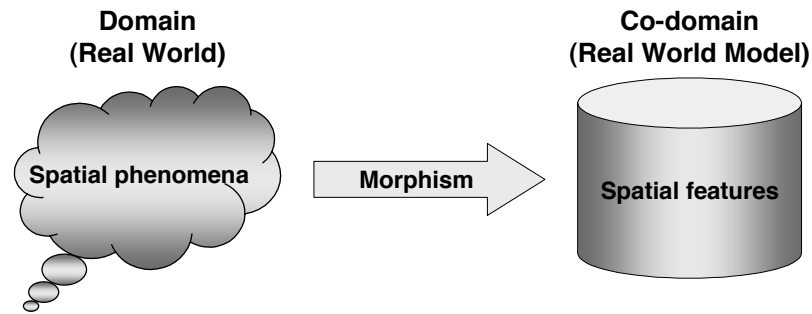


Figure 4: Spatial modeling is a structure preserving mapping from the real world to a spatial model.

As mentioned above, we consider space to be the three-dimensional Euclidean space of our common sense. All phenomena exist in this space and undergo changes, which we perceive as the passing of time. In this sense, time is modeled indirectly as changes in (spatial or thematic) attributes of the features.

We call a spatial data model that also considers time a *spatiotemporal data model*. Sometimes such a model is addressed as four-dimensional, giving the impression that time is the fourth dimension in addition to the three spatial dimensions. It is, however, better to call it spatiotemporal instead of four-dimensional. First, time is not of the same type as the spatial dimensions; it has a distinct different quality. Secondly, the term ‘four-dimensional’ only makes sense when we always would consider three spatial dimensions. In most of the cases, however, the spatial data model only considers two dimensions.

3.4 Real World Models and Their Representation

Spatial data are computer representations of spatial features. A modeling language for a GIS database is a *spatial data model*. It is used in the design of spatial databases. A spatial database holds a digital representation of the real world, sometimes called a *digital landscape model* (DLM).

A DLM is the basis for spatial analysis and manipulation of spatial data. The feature space for a spatial database is a geometric space in which we model features at various levels of detail. A good data model should allow for multiple representations of spatial features. The transformation of a representation from a detailed to a less detailed version is called *model generalization*. This is different from cartographic generalization where a graphic representation (digital or analogue) at a smaller scale is derived from a large-scale dataset or map under certain merely graphic constraints.

From the database (DLM), we derive graphic representations in digital or analog form (*cartographic model*) that can result in a cartographic product. Figure 5 shows the different aspects of data modeling for database and cartographic design. Database creation is a two-step process: first, the database is designed by defining a database schema that identifies types but no occurrences. Secondly, the database is filled with real data, thereby creating a digital landscape model.

We are particularly interested in properties of the geometric space that remain invariant under certain transformations. This is important to guarantee a consistent representation of the features in a database. These properties are related to *topology*, which will be discussed later in this chapter.

Some features possess a degree of *uncertainty* in either their thematic or spatial extent. For example, soil types do not have crisp boundaries. Linguistic expressions of vagueness or uncertainty such as “moderate slope” or “close by Enschede” are often a better analysis approach than clear cut class boundaries. We have to devise proper means to take care of these uncertainties in our spatial data models. *Fuzzy logic*, a branch of modern mathematics, provides the tools.

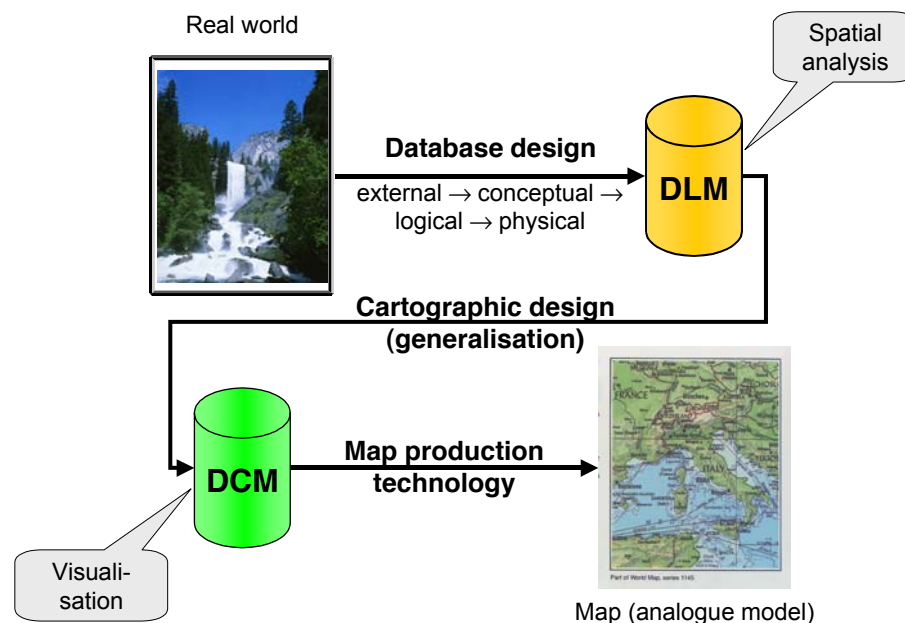


Figure 5: Data modeling from the real world to a database, and from there to digital cartographic models and analogue products for visualization.

The theory of fuzzy sets is based on the principle of a membership function that determines for every element to which degree it belongs to a set. The membership grade is a real number between zero (no membership) and one (complete membership). With a proper membership function (user-defined and depending on the application), we can model vague thematic and spatial attributes of spatial features. Spatial analysis can be performed on fuzzy sets in a similar way as on crisp sets.

Spatial data exist not only in space but also in time. They carry temporal characteristics that data models must be able to handle. Several spatiotemporal data models have been proposed. Some of them will be discussed below (see section 3.4.4).

3.4.1 Database Design

In designing a database, we have to pass through several design phases. The result of every phase is a schema of the respective level. A schema is a representation of the database as described with a particular data model. This approach is called the ANSI/SPARC architecture; it comprises four levels.

A database is generally thought to serve multiple users or user groups. They may have different perceptions of the data stored. At this level, each user (group) is supported with its own *external view* of the database. We may define an external view as a personalized conceptual database schema. There will be as many external views as there are users and user groups. Mainly domain experts do spatial modeling at this level.

All external views are merged into a single *conceptual schema* of the database. This is usually done with high-level semantic data models such as the entity-relationship model (ER model), the extended ER model (EER), or object-oriented data models. The basic constructs of the E-R model are *entity types* (e.g., country), *attribute types* (e.g., population) and *relationship types* (e.g., neighbor of). Instances of the types populate the database, e.g., ‘The Netherlands’, ‘15 million’, and ‘Germany, Belgium’ are instances of the entity type ‘country’, the attribute type ‘population’, and the relationship type ‘neighbor of’, respectively. A conceptual schema is implementation-independent and not related to any particular database management software. It provides an answer to the question what phenomena are represented in the database.

The conceptual schema is transformed into a *logical schema* using one of the logical data models. Currently, the most popular one is the relational data model, which is based on relational algebra. Most commercial database implementations provide support for this model. It is easy to understand because it is based on relations—sets of records—that have a straightforward implementation as tables. The logical schema is meant to provide the definition of a redundancy-free dataset. Each fact should be stored only once in the database.

A *physical schema* is the result of the implementation of the logical schema with particular database management software. Table 6 summarizes the ANSI/SPARC architecture.

Table 6: Data models and schemas in database design (the ANSI/SPARC architecture).

Schema	Models used to derive the schema
External views	Depending on different user perspectives, a subset of the real world is defined and described (spatial modeling).
Conceptual schema	A synthesis of external views to create a conceptual schema making use of semantic data modeling techniques such as the entity-relationship approach.
Logical schema	Transformation of the conceptual schema into a logical schema using the relational model. Emphasis is on redundancy removal.
Physical schema	This is the mapping of the logical schema into data structures and algorithms. It is normally not accessible or visible to the user. Its emphasis is on processing speed.

3.4.2 Topology and Spatial Relationships

Topology is the branch of mathematics that deals with properties of spaces that remain invariant (i.e., do not change) under certain transformations. A simple example will illustrate what we mean.

Assume you have some geometric figures that are drawn on a sheet of rubber (Figure 6). Now, take the sheet and pull on its edges, but do not tear or break it. The figures will change in shape and size. Some properties, however, do not change.

E is still inside D, the neighborhood relationships between A, B, C, D, and E remain, and the boundary lines have the same start and end points. The areas are still bounded by the same boundary lines, only their shapes and the length of the perimeters have changed.

These relationships are invariant under a continuous transformation. The properties are called topological properties, and the transformation is called a *topological mapping* (or *homeomorphism*). The result is topologically equivalent (or homeomorphic) to the original dataset.

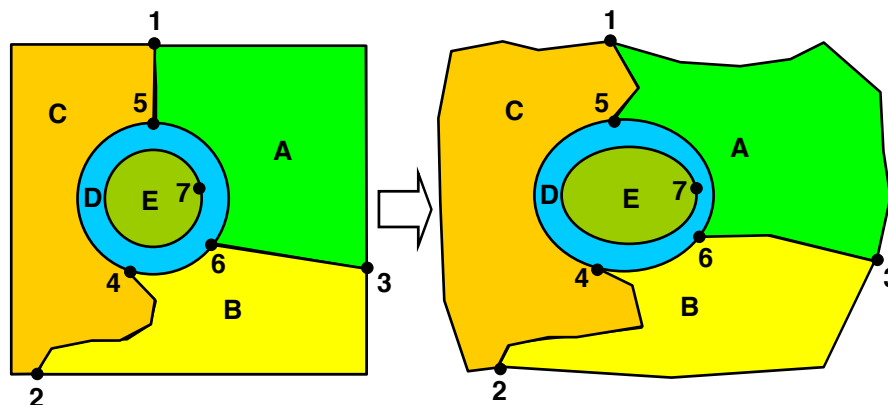


Figure 6: Rubber sheet transformation.

The mathematical properties of the geometric space used for spatial data can be described as follows:

The space is a three-dimensional *Euclidean space* where for every point we can determine its three-dimensional coordinates as a triple (x, y, z) of real numbers. In this space, we can define points, lines, polygons, and volumes as geometric primitives of the respective dimension. A point is a zero-dimensional, a line a one-dimensional, a polygon a two-dimensional, and a volume is a three-dimensional primitive.

The space is a *metric space*, i.e., we can always compute the distance between two points according to a given distance function (metric).

The space is a *topological space*, i.e., for every point there exists a neighborhood that is topologically equivalent (homeomorphic) to an interval (one-dimensional), disk (two-dimensional) or ball (three-dimensional), depending on the dimension of the space in use. These neighborhoods allow us to define open sets that under certain conditions define a topology on the space.³

We can structure space by simple topological sub-spaces that are easy to handle and can be used as building blocks for spatial data (Figure 7). They are called *simplices* and are the simplest geometric

³ A formal introduction to topological spaces is beyond the scope of this text. We refer the reader to the literature mentioned in the bibliography.

shapes of a respective dimension: point (0-simplex), line segment (1-simplex), triangle (2-simplex), and tetrahedron (3-simplex). It is common to use their homeomorphic images (or *cells*) in spatial data models. Cells are nodes (0-cells), arcs (1-cells), polygons (2-cells), and volumes (3-cells). A collection of cells (or *cell complex*) is the topological equivalent of a collection of spatial features (Figure 8).

Interior and *boundary* are properties of spatial features that remain invariant under topological mappings. This means, that under any topological mapping, the interior and the boundary of a feature remains unbroken and intact.

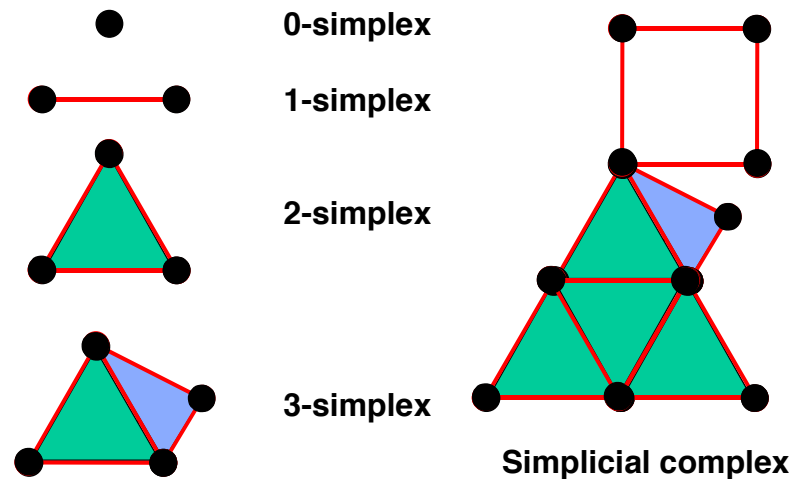


Figure 7: Simplexes and simplicial complex. Features are approximated by a set of points, line segments, triangles, and tetrahedrons.

In many applications, we will use a two-dimensional space only, i.e., the usual Euclidean plane. In fact, data models for three-dimensional spatial data are more difficult than two-dimensional data models.

The consistency of a spatial dataset is important for any processing and analysis that might be applied. The data model must provide means to define and enforce *topological consistency* in a database. Topology provides us with a set of rules that can easily be applied to any two-dimensional spatial dataset.

If we can show that the following rules are satisfied for every element in the dataset, we know that it is a topologically consistent two-dimensional configuration.

1. Every arc must be bounded by two nodes (start node and end node).
2. For every arc, there exist two polygons (left polygon and right polygon).
3. Every polygon has a closed boundary consisting of an alternating sequence of nodes and arcs.
4. Around every node, there exists an alternating closed sequence of arcs and polygons.
5. Arcs do not intersect except at nodes.

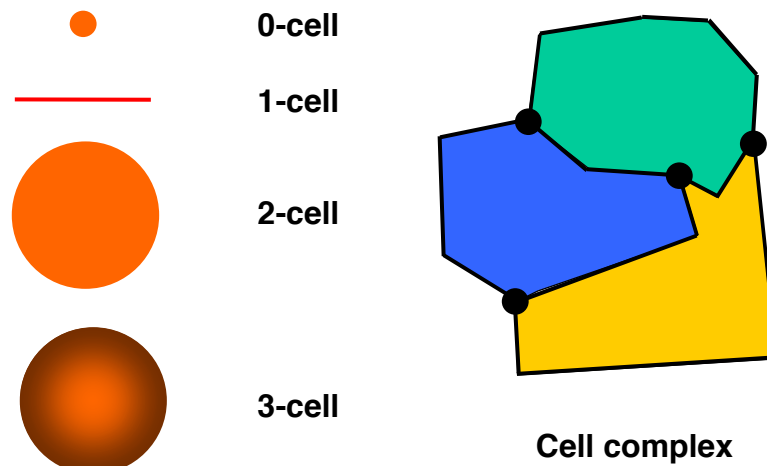


Figure 8: Cells and a cell complex. Spatial features are represented as topological images of cells.

In mathematical terms, these rules guarantee that the configuration is a two-dimensional manifold where every point has a neighborhood equivalent to an open disk. These rules are valid only for a two-dimensional cell complex. They cannot be applied without additions and modifications to higher dimensions.

Whereas relationships between simplices or cells define consistency constraints for spatial data, we can use the topological properties of interior and boundary to define relationships between spatial features. Since the properties of interior and boundary do not change under topological mappings, we can investigate their possible relations between spatial features.⁴

Let us assume two spatial regions A and B. Both have their respective boundary and interior. When we consider all possible combinations of intersections between the boundaries and the interiors of A and B, we know that these will not change under any topological transformation. From these intersection patterns, we can derive eight mutual spatial relationships between two regions. If, for instance, the boundary of A intersects the boundary of B, and the interiors of A and B do not intersect, we say that A and B meet. Figure 9 shows all possible eight spatial relationships: disjoint, meet, equal, inside, covered by, contains, covers, and overlap. These relationships can be used, for instance, in queries against a spatial database.

Networks, like road or river networks, are best modeled as a one-dimensional subset (or *skeleton*) of a cell complex. The topological relationships are then reduced to incidence relationships between edges (arcs) and nodes. Such a structure is also called a *graph*. *Graph theory*, although closely related to topology, has developed as an independent mathematical discipline.

⁴ We restrict ourselves here to relationships between spatial regions (two-dimensional features without holes).

A special type of graph frequently used in GIS is a *planar graph*. It is completely embedded in the plane such that no two edges intersect except at nodes.⁵

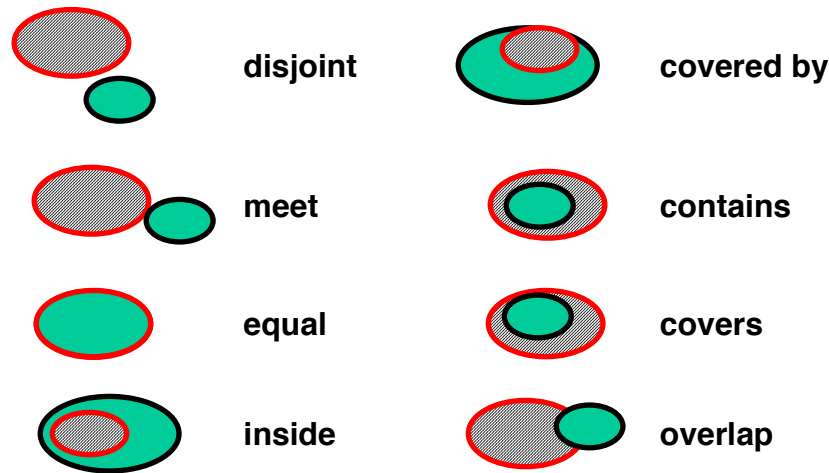


Figure 9: Spatial relationships between two regions derived from the topological invariants of boundary and interior.

3.4.3 Spatial Data Models

Among spatial data models, we can distinguish two major types, field- and object-based models. *Field-based models* consider spatial phenomena to be of a continuous nature where in every point in space a value of the field can be determined. Examples of such phenomena are temperature, barometric pressure, or elevation. *Object-based models* consider space to be populated by well distinguishable, discrete, bounded objects with the space between objects potentially being empty. Examples are a cadastre with clearly identifiable objects like parcels and buildings.

Field versus object can be viewed—in general terms—as a manifestation of the philosophical conception of *plenum* versus *atomic* space (see Section 3.2.1), or as in modern physics, *wave* versus *particle* (see Section 3.2.3). In GIS, fields are usually implemented with a *tessellation* approach, objects with a (topological) *vector* approach. The following sections briefly illustrate the two different model types.

3.4.3.1 Field-based Models

The underlying space for a field-based model is usually taken as the two- or three-dimensional Euclidean space. A *field* is a computable function from a geometrically bounded set of positions (in 2D or 3D) to some attribute domain. Computable means that for every position within the geometric bounds a value can be determined by either measurement or by computation. A *field-based model* consists of a finite collection of such fields (Figure 10).

Fields can be discrete, continuous and differentiable. Discrete fields represent features with boundaries; continuous fields are used for features where the underlying function is continuous, such as for

⁵ Graphs are the logical choice for networks. It should be noted, however, that some (historic) two-dimensional data modeling approaches have been based on planar graphs. Here, the 2-cells (polygons) bounded by 1-cells (edges) and 0-cells (nodes) are counted as “faces” of the graph.

temperature, barometric pressure or elevation. If the field function is differentiable, we can even compute the slope at every position.

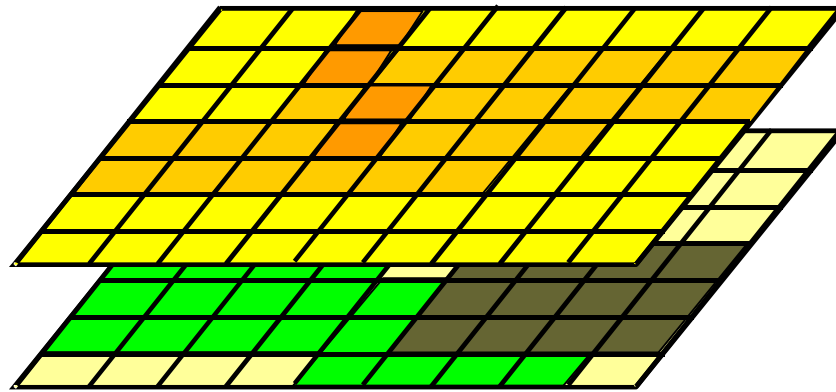


Figure 10: Two data layers in a field-based model.

It is important to note that our definition of field is completely independent of implementation considerations. This means that we do not have considered a data structure to represent them. It is, however, obvious that some data structures seem to be the logical choice for certain types of field.

Though geometrically bounded, the domain of a field is still an infinite set of positions. Computers cannot easily represent field values for all these positions, so we must accept an approximation. The standard way to obtain this is to finitely represent the geometrically bounded space through a subdivision into a regular or irregular *tessellation*, consisting either of square (cubic) or triangular (tetrahedral) parts. These individual parts are called *locations*; points often approximate them.

Our finite approximation of positions into locations leads to some forms of interpolation that must be dealt with. The field value at a location can be interpreted as one for the complete tessellation cell, in which case the field is discrete, not continuous or even differentiable. Some convention is needed to state which value prevails on cell boundaries; with square cells, this convention often says that lower and left boundaries belong to the cell.

Another option is to interpret the field value at a location as representative only for some position within the cell. Again that position is fixed by convention, and may be the cell centroid or, for instance, its left lower corner. Field values for positions other than these must be computed through some form of interpolation function, which will use one or more nearby field values to compute the value at the requested position. This allows to represent continuous, even differentiable, functions.

To represent spatial features using a field-based approach we have to perform the following steps:

- Define or use a suitable model for the underlying space (tessellation).
- Find suitable domains for the attributes.
- Sample the phenomena at the locations of the tessellation to construct the spatial field functions.
- Perform analysis, i.e., compute with the spatial field functions.

In Chapter 5, we will discuss some more details regarding tessellations, also in cases where the space is irregularly divided.

3.4.3.2 Object-based Models

Object-based models decompose the underlying space into identifiable, describable *objects* that have spatial, thematic, and temporal *attributes*, as well as *relationships* among each other (Figure 11). The space outside the objects is empty. Examples of objects are buildings, cities, towns, districts or countries; attributes are, for instance, the number of floors, population, boundary or area.

In a GIS database implementation, objects are represented as a structured collection of geometric primitives (points, lines, polygons, and volumes) under geometric, thematic and topological constraints.

Object-based models are discrete models. Operations in the model always refer to the manipulation of individual objects or sets of objects. Manipulations concern the spatial, thematic, topological or temporal domain. Accordingly, they are realized through geometric, attribute manipulation or topological operations.

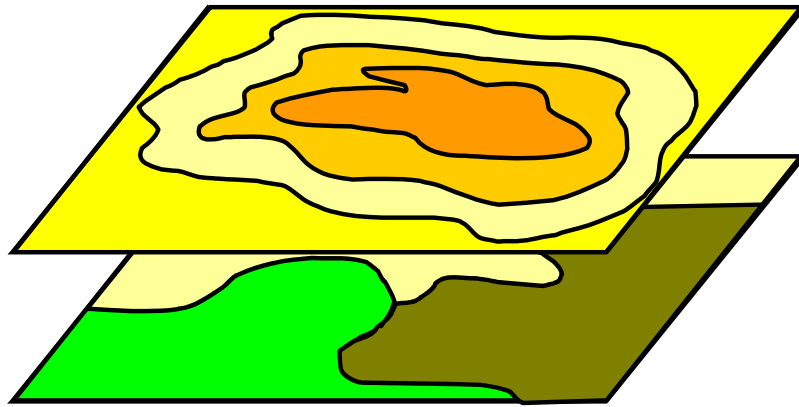


Figure 11: Layers in an object based model.

Topology plays a major role in object-based models. It is the “language” that allows us to specify and enforce consistency constraints for spatial databases.

The majority of object-based models are two-dimensional. Recently, three-dimensional data models have been proposed. Their topology is more difficult to handle than for the standard two-dimensional cases.

3.4.4 Spatiotemporal Data Models

Beside geometric, thematic and topological properties, spatial data possess also temporal characteristics. It is, for instance, interesting to know who were the owners of a land parcel in 1980, or how did land use of a given piece of land change over the last 20 years.

Spatiotemporal data models are data models that can also handle temporal information in spatial data. Several models have been proposed. The most important ones will be discussed briefly.

Before we describe the major characteristics of the spatiotemporal data models, we need a framework to describe the nature of time itself. Time can be characterized according to the following properties:

Time density. Time can be *discrete* or *continuous*. Discrete time is composed of discrete elements (seconds, minutes, hours, days, months, or years). In continuous time, for two points in time, there is always another point in between. We can also structure time by *events* (points

in time) or *states* (time intervals). When we represent states by intervals bounded by nodes (events) we can derive temporal relationships between events and states such as “before”, “overlap”, “after”, etc.

Dimensionality of time. *Valid time* (or *world time*) is the time when an event really happened. *Transaction time* (or *database time*) is the time when the event was recorded in the database.

Time order. Time can be *linear*, extending from the past to the present, and into the future. We know also *branching* time (possible scenarios from a certain point in time onwards) and *cyclic* time (repeating cycles such as seasons or days of a week).

Measures of time. A *chronon* is the shortest non-decomposable unit of time that is supported by a database (e.g., a millisecond). The life span of an object is measured by a (finite) number of chronons. *Granularity* is the precision of a time value in a database (e.g., year, month, day, second, etc.). Different applications require different granularity. In cadastral applications, time granularity can be a day; in geology mapping, time granularity is more likely in the order of thousands of years.

Time reference. Time can be represented as *absolute* (*fixed time*) or *relative* (*implied time*). Absolute time marks points on the timeline where events happen (e.g., “6 July 1999 at 11:15 p.m.”). Relative time is indicated relative to other points in time (e.g., “yesterday”, “last year”, “tomorrow,” which are all relative to “now”, or “two weeks later,” which may be relative to an arbitrary point in time.).

In spatiotemporal data models, we consider changes of spatial and thematic attributes over time. In data analysis, we can keep the *spatial domain fixed* and look only at the attribute changes over time for a given location in space. We would, for instance, be interested how land cover changed for a given location or how the land use changed for a given land parcel over time, provided its boundary did not change.

On the other hand, we can keep the *attribute domain fixed* and consider the spatial changes over time for a given thematic attribute. In this case, we could be interested to see which locations were covered by forest over a given period.

Finally, we can assume both the *spatial and attribute domain variable* and consider how an object changed over time. This actually leads to notions of object motion, and these are a subject of current research, with two of the applications being traffic control and mobile telephony. But many more applications are on the horizon: think of wildlife tracking, disease control, and weather forecasting. Here, the problem of *object identity* becomes apparent. When does a change or movement cause an object to disappear and become a new one?

In the following, we describe the major characteristics of some popular spatiotemporal models.

3.4.4.1 Space-Time Cube Model

This model is based on a two-dimensional space (spanned by the x - and y -axis) whose features are traced through time (along the z -axis) thereby creating a space-time cube. The traces of objects through time create a worm-like trajectory in the space-time cube. This model potentially allows absolute, continuous, linear, branching and cyclic time. It supports only valid time. The attribute domain is kept fixed and the spatial domain variable.

3.4.4.2 Snapshot Model

In the snapshot model, layers of the same theme are time-stamped. For every point in time that we would like to consider, we have to store a layer and assign the time to it as an attribute. We do not have any information about the events that caused different states between layers. This model is based on a linear, absolute, discrete time. It supports only valid time and multiple granularity. The spatial domain is fixed (field-based) and the attribute domain is variable.

3.4.4.3 Space-Time Composite Model

The space-time composite model starts with a two-dimensional situation (plane or layer) at a given start time. Every change of features that happens later is projected onto the initial plane and intersected with the existing features, thereby creating an incrementally built polygon mesh. Every polygon in this mesh has its attribute history stored with it. The space-time composite model is based on linear, discrete, relative time. It supports both valid and transaction time, and multiple granularity. It keeps the attribute domain fixed and the spatial domain variable.

3.4.4.4 Event-Based Model

In an event-based model, we start with an initial state and record events (changes) along the time line. Whenever a change occurs, an entry is recorded. This is a time-based model. The spatial and thematic attribute domains are secondary. The model is based on discrete, linear, relative time, supports only valid time and multiple granularity.

3.4.4.5 Spatiotemporal Object Model

This model is based on spatiotemporal objects (ST-objects) that are a complex of ST-atoms (STA). Both objects and atoms have a spatial and a temporal extent. The model is based on discrete, absolute, linear time, and supports valid and transaction time as well as multiple granularity.

3.5 Summary

Geographical information systems process spatial information. The information is derived from spatial data in a database. To sensibly work with these systems, we need models of spatial information as a framework for database design. These models address the spatial, thematic and temporal dimensions of real world phenomena. An understanding of the principle concepts of space and time rooted in philosophy, physics and mathematics is a necessary prerequisite to develop and use spatial data models.

We know two major approaches to spatial data modeling, the analogue map approach, and spatial databases. Today, the function of maps as data storage (map as a database) is increasingly taken over by spatial databases. In databases, we store representations of relevant phenomena in the real world. These representations are abstractions according to selected spatial data models. We know two fundamental approaches to spatial data modeling, field-based and object-based models. Both have their merits, advantages and disadvantages for particular applications.

Consistency is an important requirement for every model. Topology provides us with the mathematical tools to define and enforce

consistency constraints for spatial databases, and to derive a formal framework for spatial relationships among spatial objects.

Spatial data not only possess spatial and thematic attributes, but extend also into the temporal domain. A model of time for spatial information is an important ingredient for any spatial data model, thus leading to what is called spatiotemporal data models.

Spatial Data Structures

An important characterization of spatial data structures is derived from the way we look at two- or three-dimensional space, and how the phenomena that we are interested in are organized in that space. The primitives that are used to represent features in the object view comprise *point*, *line*, *area*, and *volume* items. It is often called the vector-based approach.

In the field view, the part of space relevant to us is divided up completely into regular or irregular *tessellations*—usually squares, triangles, or cubes—that define a two- or three-dimensional *raster* or *grid*. The more common name for the field view approach is the raster-based approach.

Below, we will pay most attention to the vector-based approaches, and we will discuss point data structures, line data structures, and area data structures.

4.1 Point Data

A good way to study a specific class of data structures is to address the questions when and how they are used. For point data we can distinguish the following four common types of query:

Point query: Is point (x, y) in the database?

Range query: Find all points in a given range (e.g., in a rectangle).

Boolean query: Logical composition of the above types (AND, OR, NOT).

Neighbor query: Find the n nearest neighbors of a given point.

Distance scan: Enumerate points by increasing distance from a given point.

The point data structures presented in the following sections are illustrated using a sample dataset taken from the textbook of SAMET (1990a). The dataset comprises locations of eight North American cities. For the sake of simplicity, we have chosen simplified values for their coordinates. The reader is referred to Table 7 for the coordinate values of these cities.

Table 7: Sample point dataset (after SAMET 1990a)

Name	x	y
Chicago	35	40
Mobile	50	10
Toronto	60	75
Buffalo	80	65
Denver	5	45
Omaha	25	35
Atlanta	85	15
Miami	90	5

4.1.1 Cell Method

The *cell method* divides space into equally sized cells (squares). The data structure is a two-dimensional array with one entry per cell. Each cell contains the head of a linked list, and the items in this list are the data points lying within the cell. The cell method is sometimes also known as the *fixed grid method*. Figure 12 is a cell method representation for the data of Table 7. Each cell is considered to be open on its upper and right boundaries and closed on its lower and left boundaries. Therefore, Buffalo is in the cell centered at $(90,70)$. This is a rather common convention for other data structures also.

A spatial index structure stores the entities in a collection of *buckets*. Typically, each bucket is associated with a single page in secondary memory, but more than one bucket may be associated with the same page. In some approaches, the bucket size may vary and a bucket may have more than one page associated with it. With each bucket is associated a *bucket region*, a part of space that contains all entities in its bucket. It is computationally easiest to let bucket regions be rectangles. For point data, buckets are best disjoint and together they should partition the relevant part of space.⁶ Point data organized with the cell method can be stored through point bucket techniques.

⁶ With area data, bucket regions are sometimes allowed to overlap.

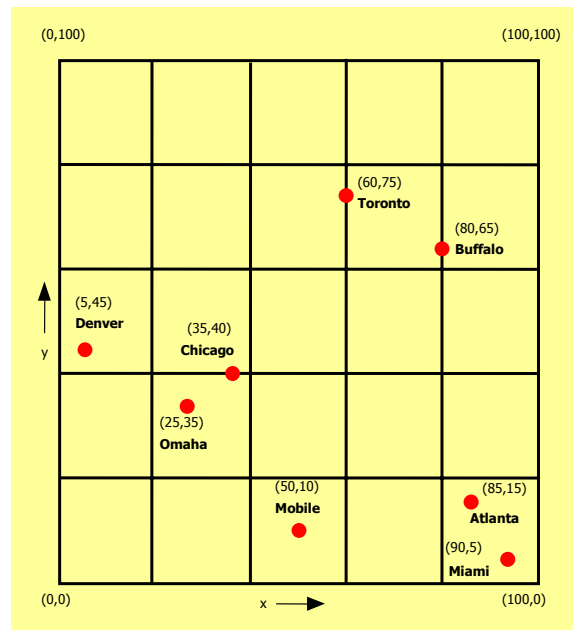


Figure 12: Grid representation for data in Table 7 (after SAMET 1990a)

4.1.2 Point Quadtrees

A point quadtree is a quaternary search tree that can be used to store point data. It is assumed that the coordinate pair of each point is unique, as otherwise overflow lists have to be maintained. A node in a point quadtree holds the data for the point, and provides four more pointers to other nodes. These pointers refer to a point in the NW, NE, SW, and SE quadrant, respectively, if such a point exists. Otherwise, the pointer is nil obviously.

This organization means that the point quadtree's shape depends on the order in which points are added to it. With reference to Figure 12, if the cities on our map are added in the order Chicago, Mobile, Toronto, Buffalo, Denver, Omaha, Atlanta, Miami, the resulting point quadtree would be as in Figure 13.

An important characteristic of navigation through a point quadtree is that at each node, if it is not the one that is searched for, the pointer to follow is determined by *two* conditions. The first is the relative position of the searched *x*-coordinate with respect to the *x*-coordinate of the current node; the other is the comparison between the *y*-coordinates of the searched and current nodes. In other words, point quadtrees require two-dimensional navigation.

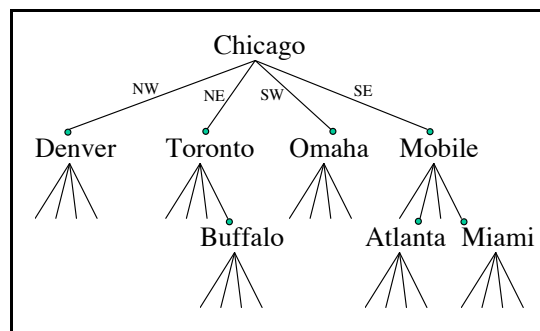


Figure 13: Point Quadtree obtained from insertion indicated in the text

4.1.3 K-D Trees

Navigation through a point quadtree can sometimes be cumbersome. For instance, when we search for a node in the tree, we have to compare both x and y coordinates at each subtree. There are a few other shortcomings of point quadtrees.

Some of these disadvantages are overcome by the k - d tree. A k - d tree in principle is a binary tree with the added requirement that at each following depth in the tree another coordinate (or other attribute) is used for navigation. In the two-dimensional case, this means that to navigate through the tree, at the root and each subsequent even depth, we use the x -coordinate value, while at the odd depths we use the y -coordinate value. This lets itself easily be generalized to arbitrary dimension k .

In the word k - d tree, k stands for the dimension of the space being represented, i.e., for our sample dataset we talk about a 2- d tree, since these points are represented in the plane.

Figure 14 shows the k - d tree for the data points in Table 7 when they are inserted in the order Chicago, Mobile, Toronto, Buffalo, Denver, Omaha, Atlanta, and Miami. Here is what happens during insertion:

1. Chicago becomes the root node (obviously).
2. Since Mobile's x -coordinate is bigger than Chicago's it goes to the right subtree.
3. Since Toronto's x -coordinate is bigger than Chicago's, and its y -coordinate is bigger than Mobile's it goes to Mobile's right subtree.
4. Since Buffalo's x -coordinate is bigger than Chicago's, its y -coordinate is bigger than Mobile's, and its x -coordinate is bigger than Toronto's, it goes to Toronto's right subtree.
5. Since Denver's x -coordinate is smaller than Chicago's, it goes to Chicago's left subtree, etc.

Like the point quadtree, the order of insertion determines the shape of the resulting tree.

Finally, we should remark that there exist alternative k - d trees that do not store real points in nodes internal to the tree, but rather only in leaf nodes. This is done by generating artificial internal nodes that help balancing the tree to some extent.

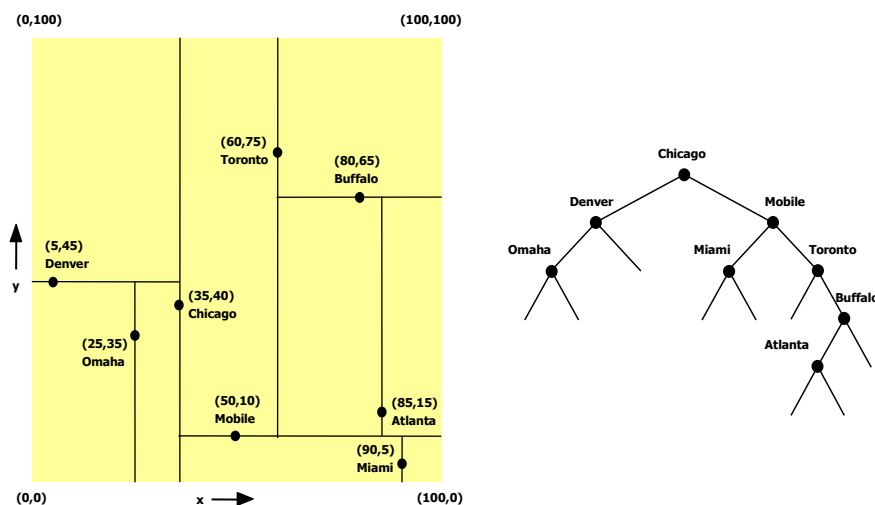


Figure 14: A k - d tree for the data in Table 7 (after SAMET 1990a)

4.1.4 PR Quadtrees

A *PR quadtree* (which stands for Point-Region quadtree) is based on a hierarchical subdivision of space into quadrants much like the standard region quadtree (for which see section 4.3.2). Like in region quadtrees, nodes are either empty, full, or intermediate. An empty node means there are no points in the quadrant of the node. A full node contains the data of a single node. An intermediate node provides a further subdivision into smaller quadrants. Figure 15 shows the PR quadtree of the data in Table 7. Every leaf node is either empty or contains a data point and its coordinates.

As with the cell method, we assume that the quadrants are on their lower and left boundaries, and open on their upper and right boundaries. The shape of the PR quadtree is independent of the order in which the data points are inserted. This is an important distinction from the point quadtree and the k-d tree.

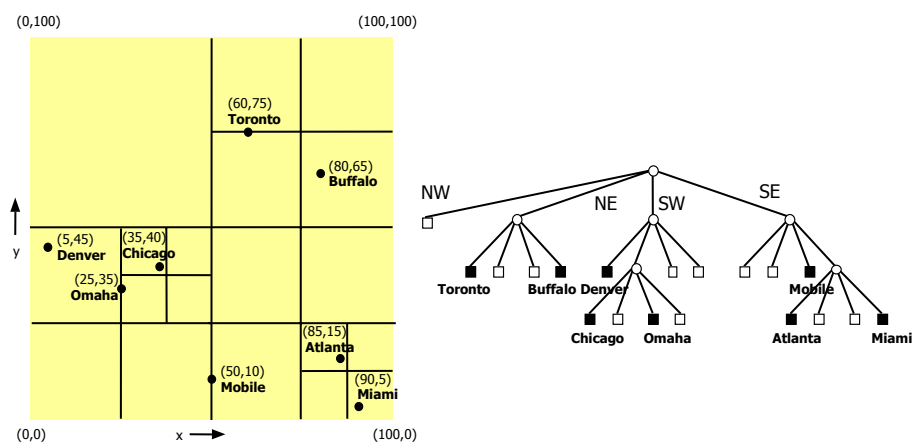


Figure 15: A PR quadtree of the data points in Table 7 (after SAMET 1990a)

4.1.5 Grid Files

The *grid file* is a 2-D or 3-D variation of the cell method that relaxes the requirement that cells are square and equal size. In grid files, cells (here called *grid blocks*) are still required to be rectangular, but that is all that is required. The purpose of grid files is to always retrieve records with at most two disk accesses, and to have efficient range query performance. A grid directory consists of grid blocks, and all points in such a block are stored in the same bucket. Several grid blocks may share the same bucket, but only if their union is again rectangular. This requirement ensures the range query performance.

The *grid directory* consists of two parts: a dynamic, two-dimensional array containing one entry for each grid block, and two one-dimensional arrays called *linear scales*. The values in the two-dimensional array are pointers to the buckets containing the data; the linear scales define a partition of the domains of the *x*-coordinate and the *y*-coordinate. These linear scales thus determine the grid block (cell) boundaries. The grid directory may be kept on disk, the linear scales are always kept in main memory. Figure 16 shows the grid file partition and the linear scales for the data of Table 7. The bucket capacity is assumed to be two, there are two different attributes (namely *x* and *y*). In this case, the grid directory consists of a two-dimensional array with nine grid blocks and six buckets labeled A to F, and two one-dimensional arrays for the linear scales. The dashed lines between the grid blocks indicate the

sharing of buckets for these blocks. Every point in a grid file can be retrieved with two disk accesses: one disk access for the grid block, which will provide us with the bucket address, and one to obtain the bucket itself.

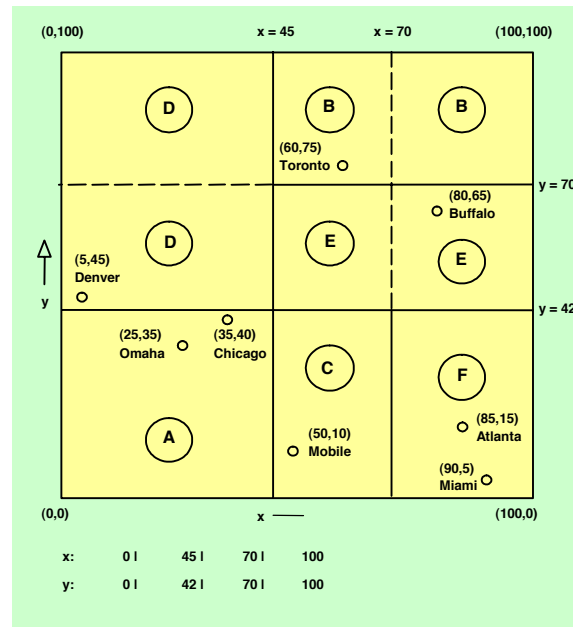


Figure 16: Grid file partition and linear scales for the data of Table 7 (after SAMET 1990a)

4.2 Line Data

Lines are the second important class of spatial data. They usually appear in two different forms:

Straight line segments. These are used to represent features like boundaries, rivers, and roads. They are frequently the result of a digitizing process.

Parameterized mathematical descriptions. These are frequently used for line data in surveying or engineering applications.

For the representation of line segments, many data structures are known. They range from simple coordinate lists to sophisticated hierarchical data structures based on quadtrees.

4.2.1 Arc/Node Structure

The *arc/node structure* is based on the topological principle of representing a graph as a set of *nodes* and a set of *edges* (or *arcs*). Every line segment has a *start* and *end node*. The internal vertices of the segments are stored as a list of coordinates defining the trajectory from the start to the end node. This list is usually implemented as a two- or three-dimensional array.

If no other topology information is represented, we obtain a simple 'spaghetti' model of coordinate lists. This model can be used for producing images, but it is less useful for more advanced spatial analysis requiring more topologic knowledge. If more topology is needed, the topological relationships can be encoded in tables. The reader may be familiar with ARC/INFO's topology tables, which are a

good illustration. Figure 17 shows a network represented by an arc/node structure.

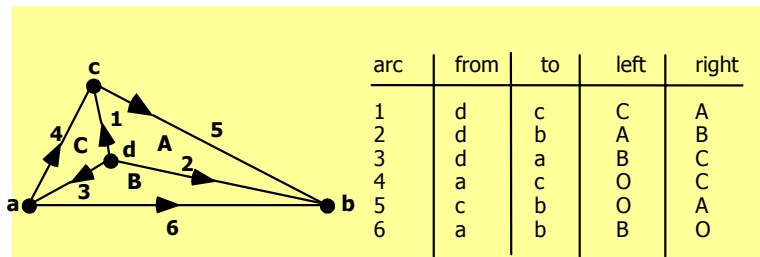


Figure 17: An arc/node structure

4.2.2 Chain Codes

A *chain code* describes a line with a sequence of unit vectors (i.e., one pixel wide) in the principal directions. A code 0123 would thus describe the boundary of a single pixel. However, it is also possible to use more than the four principal directions to allow a better approximation in case of non-rectilinear data. Figure 18 shows the codes used, and illustrates a sample line. The chain code representation of this line is $2^2 1^3 2^1 4^2 6^2 4^1$ where c^n means code c repeated n times. Chain codes are essentially raster data techniques.

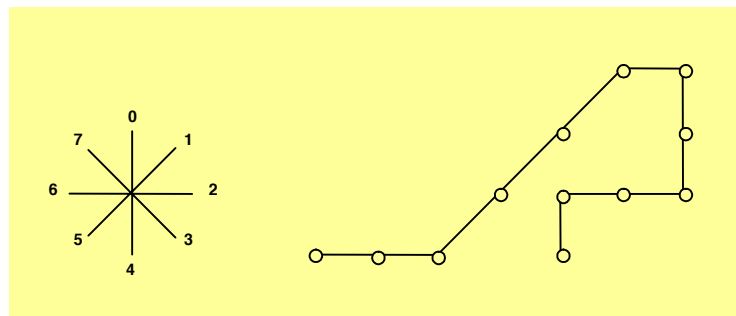


Figure 18: Chain code and sample line

4.3 Area Data

Area data can be represented on a vector basis and on a raster basis. A vector-based typically uses a boundary model, i.e., an arc/node structure for polygon boundaries and additional topological relationships for the polygons. Raster-based representation can be organized as a set of pixels with common pixel values. In this case, a region quadtree is a suitable hierarchical data structure. Such structures are not only used for the representation of data but also as a *spatial index*.

A *spatial index* is an extra data structure used to get faster access to features at a particular spatial location (point or range). It does not store those features, but provides quick navigation to them. Frequently, regions are indexed using their minimum bounding rectangles. A spatial index is built using these rectangles as address placeholders of the features. Hierarchical data structures like *quadtrees* or *R-trees* are used in this process. Spatial indices speed up the querying but slow down the updating, as the index will have to be updated also.

4.3.1 Boundary Model

The boundary model is based on the arc/node structure for polygon boundaries and additional topological relationship information stored in tables. Polygons are defined by their bounding arcs. For every arc the start node, end node, and the polygon to the left and to the right are stored. Figure 19 shows a sample set of polygons and the corresponding arc table. This approach is sometimes also called the *topological data model* or *geo-relational model*.

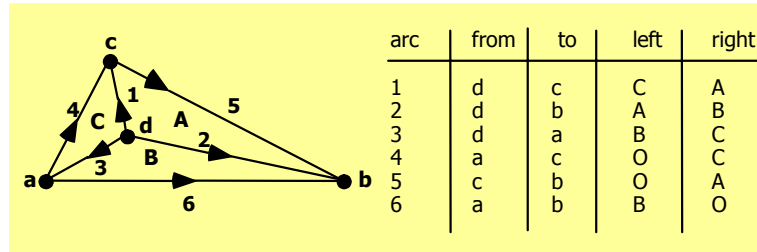


Figure 19: Sample polygons and their topological data model. The outside region (background or world polygon) is denoted as O

4.3.2 Region Quadrees

The *region quadtree* is the best known type of quadtree. It is used for two-dimensional binary regions in raster form (a pixel array consisting of only 0's and 1's). The idea is to successively subdivide a square area into four equal-sized quadrants. Quadrants are further subdivided as long as they do not contain only 0's or only 1's. A quadrant with only 0's is marked as 'empty', and a quadrant with only 1's as 'full'. The region quadtree can be characterized as a variable resolution data structure.⁷

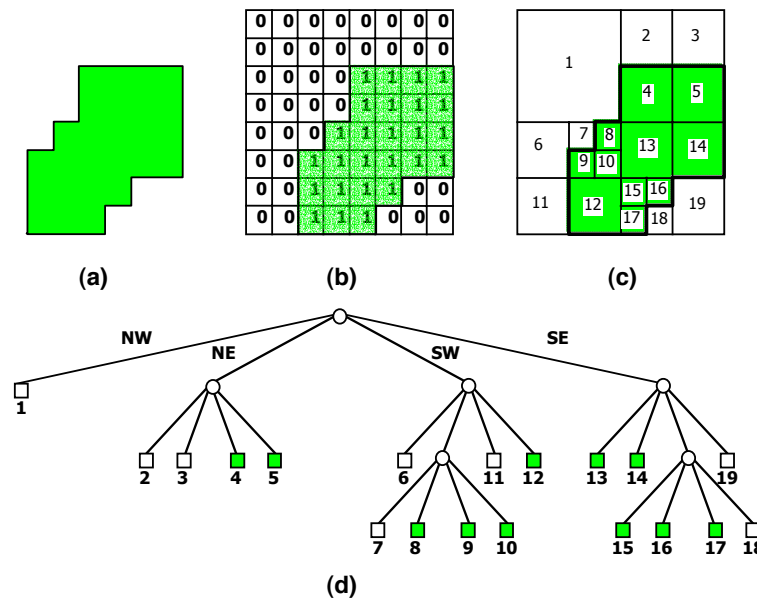


Figure 20: Sample (a) region, (b) its binary array, (c) its maximal blocks, and (d) the corresponding quadtree (after SAMET 1990a)

⁷A data structure related to the region quadtree is the pyramid (or image pyramid). Given a $2^n \times 2^n$ image array $A(n)$, a *pyramid* is a sequence of arrays $\{A(i)\}$ such that $A(i-1)$ is a version of $A(i)$ at half the scale of $A(i)$, and $A(0)$ is a single pixel. Contrary to the region quadtree the pyramid is a multiresolution data structure.

This means that a ‘full’ node in the tree stands for a completely covered quadrant, but we need to know the level of the node in the tree to decide what the size of the quadrant is. Figure 20 shows an example of a region, its binary array, its maximal blocks and the corresponding quadtree. For instance, nodes 8 and 14 both represent full quadrants, but quadrant 8 is four times smaller than quadrant 14.

To facilitate the implementation of a quadtree, an indexing method is applied making use of *locational codes* corresponding to the quadrants: 0 (NW), 1 (NE), 2 (SW), 3 (SE). This encoding is applied throughout all subquadrants in a consistent manner. It provides a linear representation of the quadtree leaf nodes (*linear quadtree*). Figure 21 shows the locational codes corresponding to the quadtree in Figure 20.

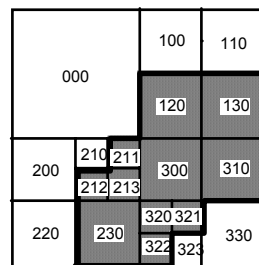


Figure 21: Locational codes corresponding to the quadtree in Figure 20 (after SAMET 1990b)

4.4 Volume Data

Volume data structures are needed for the representation of three-dimensional features. In analogy with two-dimensional representations of regions, data structures for volume data can be designed.

A straightforward extension of the region quadtree is the *octree* where, instead of two-dimensional pixels, three-dimensional volume elements (or *voxels*) are included. Space is successively divided into octants until one octant contains only voxels of one kind. According to the decomposition of space into octants a tree with eight children per node should be used.

The extension of the *boundary model* to three dimensions leads to a topological description of solid objects by their bounding faces (nodes, arcs, polygons) and the topological relationships among them. Tabular techniques for topological relationships are far from trivial.

4.5 Classification of Spatial Data Structures

Data structures can be classified according to the following criteria:

Data structures that are mainly used for representation in main memory (or *in-core* applications) and those that are used for efficient access to disk data (*bucket* methods). The total volume of the data is the discriminating factor here.

Hierarchical and *non-hierarchical* data structures: hierarchical data structures are based on a recursive (regular or irregular) decomposition of space (e.g., squares, or triangles). Hierarchically organized data are usually relatively easy to search through, and insert into by using recursive algorithms.

Data types can be based on *vector* and *raster* techniques. The determining factor quite often is the procedure followed for data acquisition. The need for spatial analysis techniques is another important factor.

Table 8 gives an overview of the data structures presented here according to their distinguishing criteria.

Table 8: Classification of data structures

Data Structure	In-core	Bucket	Hierar- chical	Non- Hierar- chical	Vector	Raster
cell method		✓		✓	✓	
k-d tree	✓		✓		✓	
point quadtree	✓		✓		✓	
PR quadtree	✓		✓		✓	
grid file		✓		✓	✓	
arc/node structure	✓			✓	✓	
chain code	✓			✓	✓	✓
boundary model	✓			✓	✓	
region quadtree	✓		✓			✓
octree	✓		✓			✓

Spatial Analysis

The most distinguishing part of a GIS are its functions for spatial analysis, i.e., to process given data in order to derive new information. Spatial queries and process models play an important role in satisfying user needs. The combination of a database, GIS software, rules, and a reasoning mechanism (implemented as a so-called inference engine) leads to what is known as spatial decision support systems (SDSS).

This chapter introduces the mentioned concepts and lists the most common analytical functions that a GIS should support. Here, we do not discuss remote sensing and image processing, although today both GIS and remote sensing are integrated to a large extent and support spatial data handling in a complementary fashion.

5.1 Data Organization and Spatial Queries

A GIS differs from other information systems by its ability for *spatial analysis*. In a GIS data are stored in *layers* (or *themes*), e.g., hydrology, transportation, land use. Usually, several themes are part of a *project*.⁸ The functions of a GIS for spatial analysis use the spatial and non-spatial attributes of the data in a spatial database to answer questions about the real world.

Models play an important role. They can be described verbally, as a mathematical function or as a set of spatial relationships stored in the form of maps or a spatial database. In the context of spatial analysis models mainly refer to process models, i.e., they are a representation of a natural or man-made process. The kind of questions, answers and the respective GIS functions are listed in Table 9.

Table 9: Types of queries

Questions	Answers	GIS functions
What is...?	Display of data as maps, reports and tables, e.g., What are the name and the address of the owner of that parcel of land?	Storage and query functions
What pattern...?	A pattern in the data, e.g., all parcels with an area greater than 2000 m^2	Query functions with constraints
What ... if...?	A prediction about the data at a certain time or at a certain location	Modeling functions

The following presents the most important query and analysis functions of a GIS (after ARONOFF 1989). They are grouped into three major classes according to common characteristics.

5.2 Maintenance and Analysis of the Spatial Data

Representatives of this class mainly operate on the spatial attributes of GIS data, and provide a user with functions as described below.

Format transformations functions convert between data formats of different systems or representations, e.g., reading a DXF file into a GIS.

Geometric transformations assist in setting up an original for digitizing. They transform device coordinates (coordinates from digitizing tablets or screen coordinates) into world coordinates (geographic coordinates, meters, etc.).

Map projections provide means to map spatial geographic coordinates onto a flat surface (map) and vice-versa.

Edge matching is the process of joining two or more map sheets. At the map sheet boundaries feature representations have to be matched to each other.

⁸ In ARC/INFO, for instance, a *workspace* corresponds roughly to a project, and *coverage* to a layer.

Editing of graphic elements. Features that have been digitized need to be edited in order to remove and correct errors, and to prepare a clean dataset for topology building.

Coordinate thinning is a process that often is applied to remove redundant coordinates from line representations. This is related to the cartographic generalization function of (line) simplification (generalization).

5.3 Maintenance and Analysis of the Attribute Data

These functions operate on the non-spatial attributes of the data. They correspond mainly to functions of conventional database systems and contain attribute editing and attribute query functions.

5.4 Integrated Analysis of Spatial and Attribute Data

Functions of this class operate on both spatial and non-spatial attributes of data, and can be grouped into the following subclasses.

5.4.1 Retrieval, classification, and measurement

Retrieval functions allow the selective search and manipulation of data without the need to create new entities.

Classification allows to assign certain features to classes according to their attributes or attribute ranges (definition of data patterns).

Generalization⁹ is a function that joins sub-classes with common characteristics to a higher level (generalized) class¹⁰.

Measurement functions allow to measure distances, length, or areas.

5.4.2 Overlay Operations

Overlay operations belong to the most frequently used functions in a GIS application. They allow to combine two different layers and apply the set theoretic operations of intersection, union, difference, and complement.

5.4.3 Neighborhood Operations

Neighborhood functions operate on the neighboring features of a given feature or set of features.

Search functions allow the retrieval of features that fall within a given search window (rectangle, circle, or polygon).

Line-in-polygon and point-in-polygon functions determine whether a given linear or point feature is located within a given polygon, or they report the polygon(s) that a given point or line are contained in.

⁹ The DISSOLVE function in ARC/INFO belongs to this type.

¹⁰ The term *generalization* has different meanings in different contexts. In geography often the term aggregation is used to indicate a process that we called generalization. In cartography generalization means either the process of producing a graphic representation of smaller scale from a larger scale original (*cartographic generalization*), or the process of deriving a smaller resolution representation from a more detailed representation within a database (*model generalization*). Finally, in computer science generalization is one *abstraction mechanism* in object orientation.

Topographic functions compute the slope or aspect from a given digital representation of the terrain (digital terrain model or DTM).

Interpolation functions predict unknown values using the known values at neighboring locations.

Contour generation functions calculate contours as a set of lines that connect points with the same attribute value. Examples are points with the same elevation (contours), depth (bathymetric contours), barometric pressure (isobars), or temperature (isothermal lines).

5.4.4 Connectivity Operations

The characteristic of these operations is that they accumulate values as they traverse a feature or a set of features.

Contiguity measures evaluate characteristics of spatial units that are contiguous (are connected with unbroken adjacency). An example would be the search for a contiguous piece of forest of a certain area and shape.

Proximity functions. The best known example of a proximity function is the buffer zone generation (or buffering).

Network analysis is used to computer the shortest path (in terms of distance or travel time) between two points in a network (*routing*), or all the points that can be reached within a given distance or duration from a center (*allocation*).

Visibility functions are used to compute the points that are visible from a given location (viewshed modeling or viewshed mapping) from a digital terrain model.

Technical Infrastructure

Every GIS needs an infrastructure that enables it to function. Such an infrastructure can be regarded as a technical and organizational infrastructure. The technical infrastructure mainly deals with the hardware and network architectures. The organizational infrastructure addresses issues of procedures and policies to make GIS work.

Information processing systems are computer systems with appropriate hardware for the processing, storage and transfer of data, as well as software components for the management of the hardware, peripheral devices and data. This chapter discusses the components of an information processing system with special consideration for spatial data handling and information processing.

First, a brief introduction to computer hardware and software is given with a description of the peripheral devices used for the capture and display of spatial data. Software components, computer networks, the Internet, and the World Wide Web are discussed.

6.1 Introduction

Geo-information infrastructures only function effectively when reliable and efficient computing and communication technologies are in place. These technologies must, in particular, support communication among computer systems and networks. Increasingly large databases have to be linked for the transfer of data or to provide a basis for interoperability of heterogeneous software and hardware systems. Many of these technologies have resulted from military research and applications. The Internet is perhaps the best known example. Together with the World Wide Web they currently form the most widely used vehicles for geo-information infrastructures.

6.2 Basic Hardware And Software

6.2.1 Computer Hardware

Computer hardware is a necessary component of an information processing system. Electronic computers exist only since about 60 years. Within this short period, we saw a rapid development that we had never seen before in modern technology. When we compare the size, performance and price of a computer in the early 1980's with the systems available today, we see orders of magnitude in performance increase and an equally steep decrease in size and price.

A computer is a general-purpose machine that can be programmed to manipulate symbols. Computers can perform calculations precisely, reliably and at high speed on large volumes of data. When we look at the architecture of a general-purpose computer, we can distinguish the following three components: a central processing unit (CPU), memory (storage), and peripherals (see Figure 22).

The top half is the central processing unit and its directly accessible parts. The bottom half indicates various parts that may have been added, such as external memory and other devices as keyboard, mouse, printer, and modem. In between the two are interfaces that enable the communication; they consist of hardware cards and cabling, and conventions about what communication signals mean.

The *central processing unit* (CPU) is the part of a computer that controls all other parts. It consists of the control unit, the arithmetic and logic unit (ALU), and memory (registers, cache and RAM).

The control unit fetches instructions from memory, decodes them, and produces signals to control other parts of the computer. The arithmetic and logic unit performs arithmetic and logic calculations (addition, multiplication, and logical operations like AND and OR). A separate floating-point unit (FPU) often performs floating-point—i.e., arithmetic with real numbers—instructions.

The CPU uses different types of memory. Registers are a small set of fast memory locations that are used by the ALU (or FPU), while a cache is fast intermediate memory that holds data that have been recently accessed, thereby speeding up subsequent access to the same data. Random access memory (RAM) is the main memory that holds the data and programs to be executed. The RAM size of modern personal computers is minimally 128 Mbytes (megabytes; 1Mbyte = 2^{10} = 1024 kilobytes, where a kilobyte is 1024 bytes) or more. Systems for spatial

data processing typically have 512 Mbytes (sometimes even more) as minimum requirement.

The performance of a processor is determined by cycles per second (or clock rate); in each cycle, a computer performs its most basic operations. Each instruction (like adding two numbers) takes a small number of basic operations. The clock rate is given as a frequency in megahertz (MHz; 1 MHz = $10^6 = 1,000,000$ Hz).¹¹ For instance, the first PC from the beginning of the 1980's had a clock rate of 4.77 MHz (nearly 5 million cycles per second). Today's processors, several generations later, perform at a rate of about 1 GHz and higher.

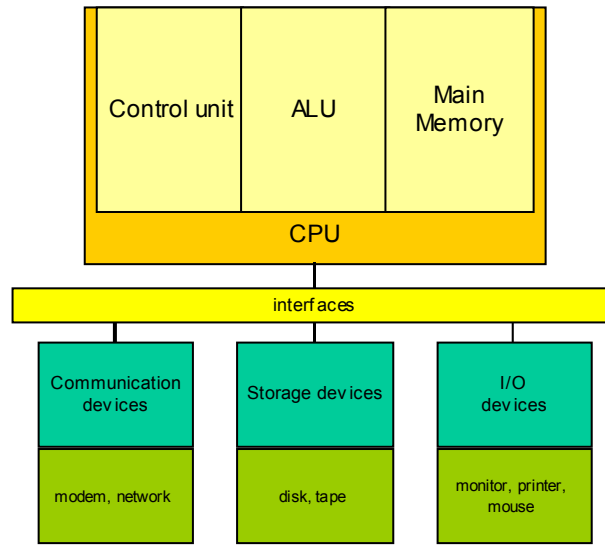


Figure 22: General computer architecture

The data are stored permanently on *storage devices*, such as disks, tapes and other magnetic or optical media. The most common devices are disks—for PC's usually with a capacity of 40 Gbytes (gigabytes; 1 Gbyte = 2^{10} Mbytes) and more. Portable disks such as floppy disks (with a capacity of 1.4 Mbytes) or other types (with a capacity of 250 Mbytes up to 2 Gbyte) allow easy transportation of relatively large datasets. Compact disks (CD's) are optical media that can be read and written using laser technology. They typically can store up to 600 Mbytes of data. Digital versatile disks (DVD) become more and more also the medium for data storage.

Communication devices link computers to networks (such as the Internet) and allow them to retrieve or exchange data over long distances without the need for physical storage media. Modems are used to establish connections over (analogue or digital) telephone lines; network cards link computers to high-speed networks (Intranet and Internet).

6.2.2 Peripheral Devices

Peripheral devices are connected to a computer and serve the task of data input and output. For a general-purpose computer we distinguish

¹¹ The Hertz is the unit of counting per second. A seconds arm on a clock ticks at 1 Hz, the minutes arm at $\frac{1}{60}$ Hz.

between the following standard input and output devices: *keyboard*, *mouse*, *monitor*, and *printer*. For spatial data, additional devices for input and output are being used: *digitizer*, *scanner*, and *plotter*.

Peripheral devices can be classified according to their purpose with regard to data flow into *input* and *output devices*. Output devices can further be distinguished into *softcopy* and *hardcopy* devices. Softcopy devices (like the computer monitor) display a non-permanent image that disappears when the screen is being erased or the power is turned off. Hardcopy devices (e.g., printers) produce a permanent display on a physical medium like paper or film.

The CPU controls peripheral devices. They are linked to the CPU through interfaces. These are controller cards for disk and tape drives, graphic and video cards for display devices, network cards and modems for communication devices, and general-purpose interface cards for serial and parallel communication with for instance a printer, plotter, scanner, mouse or digitizer.

Keyboard. A keyboard is used to enter text (characters and numbers) and to control the operating system of the computer. The keyboard layout differs among natural languages and many users need some time to adjust to a keyboard when they use computer systems different from the one used in their home country. Beside keys for characters of the alphabet and punctuation marks (!, ;, ?, etc.), every keyboard contains also function and control keys.

Mouse. Another common input device is the mouse. This is a pointing device with one, two or three buttons used to identify objects on the monitor screen. A mouse is not only used to control objects but also to enter graphic primitives (lines, shapes, and symbols) into a software application. The movement of the mouse over a flat surface—usually on a mouse pad—is echoed by a corresponding movement of a cursor—usually an arrow, cross hair, or other symbol—on the screen.

Monitor. The computer monitor is a graphic output device. Today's systems all use color graphics monitors with high resolution, based on cathode ray tube (CRT) or liquid crystal display (LCD) technology. LCD displays are flat-panel displays, and we distinguish between active and passive matrix displays. Active-matrix displays (like the Thin-Film Transistor or TFT display) produce a brighter and sharper image than passive-matrix displays. They are, however, more expensive. The graphic image on the monitor screen is represented by a raster image where every raster cell has a color code assigned. For instance, to be able to display 256 (= 2^8) different colors per raster cell (picture element or pixel) we need 1 byte of storage space per pixel.

The resolution of a monitor is measured in the number of dots that fill the monitor screen in horizontal and vertical direction. Typical resolutions are 640 (horizontal) by 480 (vertical), 800 by 600, 1,024 by 768, and 1,280 by 1,024. The more dots are used, the higher the resolution of the screen, and the sharper the image.

The chosen resolution and the available display memory of the computer system determine the number of colors that can simultaneously be displayed on the screen. With a given amount of display memory, the number of simultaneously displayable colors is indirectly proportional to the resolution, i.e., the higher the resolution the less colors can be supported. Typical numbers of colors range from 256 (hundreds) through 65,536 (thousands) up to 16,777,216 (millions).

To allow dynamic images on the screen, the raster image has to be re-drawn (refreshed) at least 24 times per second (24 Hz) to avoid flickering and adverse graphic effects. Common refresh rates are 60 Hz, 75 Hz, and 85 Hz or more.

A typical setting for a computer screen is a resolution of $1,024 \times 768$ with 65,536 simultaneously displayable colors and a refresh rate of 75 Hertz. This requires a minimum display memory size of $1,024 \times 768 \times 16$ bits equaling 1.5 Mbytes.

Digitizer. Graphic input devices frequently used for spatial data are the digitizer (or digitizing tablet) and scanners. A digitizer is a manual graphic input device that allows capturing coordinates from given source material mounted on the surface of a tablet. A pointing device (called *cursor*) is moved over the source material to capture coordinates. This cursor is equipped with crosshairs for positioning and buttons that usually have (software-defined) functions assigned to them. Whenever a button is pressed, a coordinate pair (in device coordinates as centimeters, millimeters, or inches) is captured and sent to the computer. The application program reads the coordinates and performs proper actions. This is called *point-mode digitizing*. It is also possible that the digitizer sends a continuous stream of coordinates once a button on the cursor is pressed or the cursor is moved a certain distance. We call this *stream-mode digitizing*.

The resolution of a digitizer is measured in the smallest interval of cursor movement that can be recorded. For most devices, this is 0.001 inch (or 0.0254 mm). The overall achievable positioning accuracy in manual digitizing is about 0.1 to 0.15 millimeters.

Scanner. Scanners are used for automatic digitizing. A laser beam sweeps (scans) the surface of the source document and the reflectance of light is recorded as a raster image. The result of a scan process is always a raster image with black-and-white (monochrome) or grayscale pixels according to color components (color scanner). We distinguish between flatbed scanners and drum scanners. On a flatbed scanner, the source material is placed on a flat surface and the scan head is moved over this surface to record the image. These scanners can handle source materials up to A3 format (420 by 297 mm). Drum scanners can accommodate larger source documents that are mounted on a rotating drum.

The resolution of a scanner is measured in dots per inch (dpi). Office scanners typically have resolutions between 300 and 800 dpi, i.e., about 12 to 31 dots per millimeter. Cartographic scanners for high-resolution data capture work at resolutions of 2,000 dpi (or 79 dots per millimeter).

Printer and Plotter. Modern printers serve two purposes. They are used to produce text documents and graphic representations (plots) as well. Printers make use of different technologies to produce output on the print medium. The most common are laser and ink-jet technologies. *Laser printers* transfer toner to paper (or transparency) that has been charged using a laser beam. The output can be monochrome or in color. The output format is A4 or A3. Laser printers work at resolutions of 300, 600, and 1,200 dpi.

Ink-jet printers transfer (liquid or solid) ink onto the output medium. They support different media (paper, film or transparencies) and formats (A4 up to A0). Their resolution varies between 300 and 2400 dpi. Laser printers and ink-jet printers produce a matrix of dots (a raster image) on the output medium.

Plotters are devices that are solely used for the output of graphics. Pen plotters are seldom used nowadays. They are devices that draw the graphics with a pen (or several color pens) on the output medium, producing (colored) lines. This technology is rather slow and lacks the capability to produce solid area fills. Today, pen plotters are hardly ever used.

An even less used technology is *electrostatic plotters*. These devices produce the output on specially coated paper that is electrostatically charged and subsequently moved through a toner liquid.

For high quality graphic production, *film writers* are used whose output is a film (or color separates on film) that can be used for printing. The resolution of film writers is in the range of 2,000 to 2,500 dpi. Table 10 summarizes the characteristics of the various peripheral devices.

Table 10: Peripheral input and output devices overview

Device	Resolution	Comments
pen plotter	0.001 inch	no solid color fill
ink jet plotter	300 to 2,400 dpi	
electrostatic plotter	400 dpi	
laser printer	300 to 1,200 dpi	DTP quality
film writer	2,500 dpi	cartographic quality
monitor	640 × 480 dots	VGA (Video Graphics Array)
	800 × 600 dots	SVGA (Super Video Graphics Array)
	1,024 × 768 dots	XVGA (eXtended Video Graphics Array)
	1,280 × 1,024 dots	
digitizer	0.001 inch	overall achievable accuracy is 0.1 to 0.15 mm
scanner	300 to 2,400 dpi	office scanner
	2,400 dpi	cartographic scanner

6.2.3 Software

With software, we mean computer programs that the computer hardware can execute to serve various purposes. A computer program is a list of instructions that we want the computer to carry out. It is called *software* because in contrast to *hardware*, we can easily change it—it is non-physical. We can distinguish the following types of software:

- Operating systems
- Programming languages
- Application programs

The task and purpose of an *operating system* (OS) is to control and manage the hardware components and system resources of a computer and to serve as an interface between the hardware and the users. Currently, the most popular operating systems are Microsoft Windows (Windows 98, Windows NT, Windows 2000, and Windows XP) and Unix (in various dialects such as Linux). Today, most operating systems manage multiple users. A user communicates with the OS either through a graphic user interface or by typing commands. Other, less used, operating systems are DOS (Disk Operating System, the predecessor of Windows) and systems that are mainly used for large multipurpose mainframe computers.

Application programs are written in a *programming language*, a language that allows to express instructions for the computer. We know

many languages and their dialects. The best-known and widely used are *Basic* and *C*, and object-oriented versions like *C++*. *Java* is used for Internet applications mostly, but it is in fact also a general purpose programming language. Often an operating system can be customized to serve particular user needs. Today, visual programming languages such as *Visual Basic* or *Visual C* are widely used.

Application programs perform certain tasks that are needed by users. Text processors, spreadsheets, presentation and graphics applications, database software, and programs written by end-users belong to this group. The software used for spatial data handling (image processing software, geographic information systems) is all application programs as well. Many of these have already been developed, and this means that many users do not have to be capable of programming.

6.2.4 Computer Networks

Networks allow computers to transfer data, exchange and execute programs across computers at different locations, and to acquire information made available by other users and content providers from various sources. The Internet is the most widely used computer network in the world. Millions of computers are linked to it. Data and programs are transported over the network according to defined protocols such as TCP/IP (transmission control protocol/internet protocol), FTP (file transfer protocol), and HTTP (hypertext transfer protocol).

The World Wide Web (WWW or the Web) is one of the most popular applications running on the Internet. Users communicate with the WWW through browsers (such as Netscape Navigator or Microsoft Internet Explorer).

In a network, computers have different roles. Often, one or more computers (the *servers*) serve many others with programs or data. The others are *clients* of the servers. This concept is called *client-server* architecture (Figure 23). It is an approach frequently used in modern computer networks. A computer network connects clients and servers. Individual computers are connected to the network either through permanent network links or through dial-up lines (using modems).

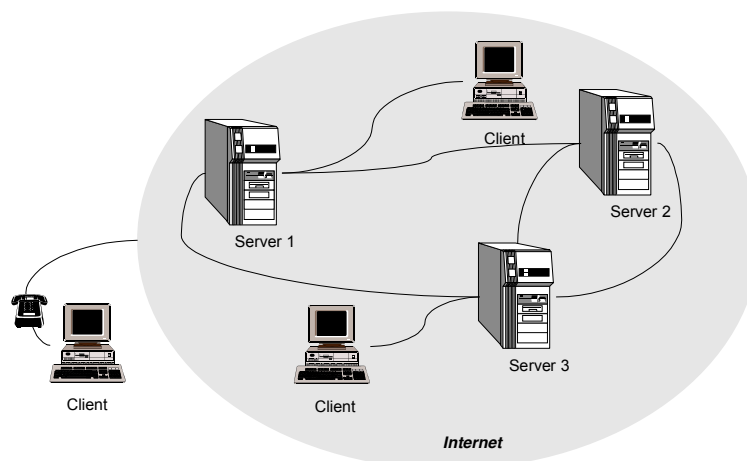


Figure 23: Client/Server concept in computer networks

6.2.5 Trends

The development of computer hardware proceeds at an enormously fast speed. Almost every six months, a faster, more powerful processor generation replaces the previous one. Software develops somewhat slower and often cannot fully use the possibilities offered by the hardware. Generally, we can observe the following trends:

- Miniaturization and improved portability of computer systems
- Increase in storage capacity and storage requirement
- Open systems and interoperability

Computers get smaller and at the same time, their performance increases. The power that we have available in today's portable notebook computers is a multiple of the performance that the first PC had when it was introduced in the early 1980's. In fact, current PC systems have orders of magnitude more memory and storage than the so-called minicomputers of 20 years ago. Moreover, they fit on an office desk. However, software providers produce application programs and operating systems that consume more and more memory. To efficiently run a computer with Windows XP and some general-purpose office applications, a PC should be minimally equipped with 256 Mbytes of main memory and 40 or more Gbytes of disk storage.

At the same time, computers have become increasingly portable. Notebook and handheld computers are now commonplace in business and personal use. Mobile phones are frequently used to communicate with computers and the Internet (through the *wireless application protocol* or WAP). The communication between portable computers and networks is still rather slow when they are connected via a mobile phone. The transmission rate currently supported by mobile communication providers is only 9,600 bits per second (bps). Digital telephone links (ISDN) support 64,000 bps up to 128,000 bps, and high-speed computer networks have a capacity of several million bps. In private households broadband Internet connections are used frequently (ADSL or Asynchronous Digital Subscriber Line).

In the future high-speed mobile communication will be available with the new standard UMTS (Universal Mobile Telecommunications System) that will allow up to 2 Mbps (Mbits/sec).

Open systems use standard architectures and protocols for networking. Interoperability is the ability of hardware and software of multiple computers from different vendors to communicate with each other. An interoperable database would for instance allow heterogeneous databases to appear as a single homogenous database to a user.

6.3 System architecture

There are several ways to organize computers so that they can perform their tasks properly. Computer systems may be arranged as individual stand-alone machines, or cooperate in computer networks. Today, computer networks are an indispensable component of any geo-information infrastructure. Special attention is given to the client-server architecture, which is of great importance not only in spatial data handling but also in general data processing.

Two important underlying principles of system architectures are open systems architectures and interoperability. *Open systems* cooperate with each other using standards for access, processing, and transfer of

data. They are not hindered by system-specific structures that impede cooperation. *Interoperability* is the ability of software and hardware on multiple computers from multiple vendors to communicate. Interoperable databases play an important role in geo-information infrastructures.

Individual computers have been used since the beginning of electronic data processing. They are usually dedicated to specific tasks or applications. However, they are not linked to anything else, and data can only be transferred this can only happen by physically transporting files on magnetic or optical media such as tapes, disks or CDs (Figure 24).

Advantages of stand-alone computers are that we know who is precisely responsible for them and that they are 'isolated' from each other, which makes unwanted transfers difficult. However, the isolation is also a drawback that causes overhead, redundancy, and frequent duplication. This can be overcome by cooperative computers in various kinds of networks or distributed systems.

A *client-server architecture* is a common form of distributed system where software is split between client and server tasks. In a distributed system, a user does not see which computer is performing which functions. It essentially behaves as a virtual single processor. The characteristic feature of a distributed system is that it is a software system running on top of a computer network.

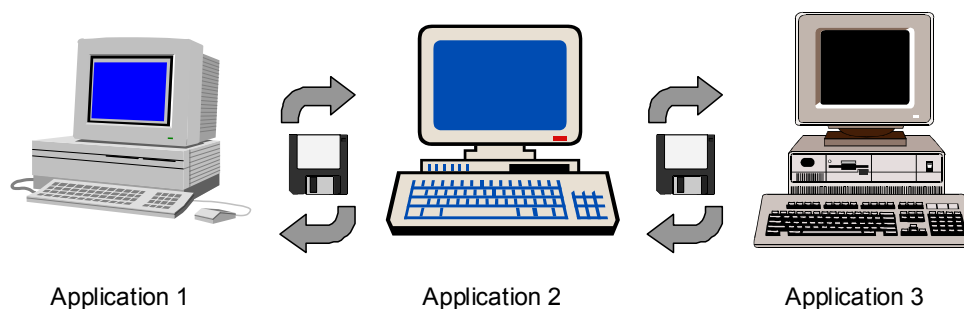


Figure 24: Stand-alone computers without cooperation

A client is a process or computer that requests the services of another computer or process (server) using some rules (protocol). The server processes the request and sends the result back to the client. The exchange of information is done through message exchange according to the protocol. A server program may run continuously, waiting for requests to arrive, or it may be invoked by some higher level program. Server programs are also called daemons.

A server computer provides services to client machines. A common example is a file server, which is a computer with a local disk servicing requests from remote clients (workstations or PCs) to read and write files on that disk. File servers often use the Network File System (NFS) protocol for workstations or Novell Netware for personal computers. In a client-server setup, we usually find many clients using a few servers. Clients and servers may even run different operating systems, and servers can be clients of other servers. Figure 25 illustrates the client-server architecture.

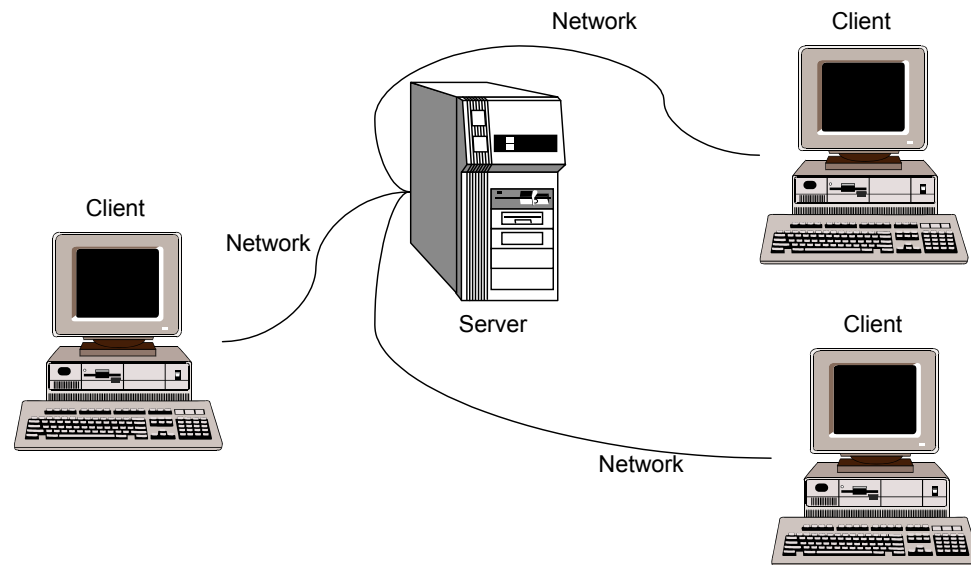


Figure 25. Client-server architecture

6.4 Computer networks

The exchange of data and information among computers in a geo-information infrastructure is a necessity. A computer network is an interconnected collection of autonomous computers. By autonomous we mean that there is no hierarchical relationship between the computers in a network. If one computer (master) is able to start or stop the others (slaves) we would call it a master-slave architecture rather than a computer network. There are various ways to connect computers in a network. The most common methods are wires, fiber optics, microwaves, and communication satellites.

The communication units that are sent between computers in a network are called messages. They can be transmitted one piece at a time from a sending to a receiving computer. If this is the case, we speak about message switching. Often it is necessary to split the message into several data packets to avoid transmission problems. If a network applies this technique we speak about packet switching. One of the principles of packet switching is that individual packets may travel distinct routes to their destination. The speed of transmission is measured in bits per second (or bps). The orders of magnitude used throughout section 3.1 are k (kilo or 10^3), M (mega or 10^6) and G (giga or 10^9). Today, transmission speeds of several hundred megabits per second are common.

6.4.1 Transmission media

The messages transmitted in a computer network are made up of bit streams. They are sent from one computer to another using different media. We can distinguish between guided media, like cables and fiber optics, and unguided media, such as radio or microwaves, and lasers. Each of these media has its own characteristics in terms of bandwidth, costs, installation, and maintenance. In the following we take a closer look at some of them.

6.4.1.1 Twisted pair

The most common medium is the twisted pair which consists of two insulated copper wires twisted together to reduce electrical interference

from other wires nearby. Each of the wires is about 1 mm thick. The twisted pair is most commonly found in telephone systems and can reach a bandwidth of several megabits per second at short distances. These cables can run several kilometers without amplification, and are usually bundled together. They are often referred to as UTP (or Unshielded Twisted Pair).

6.4.1.2 Coaxial cable

The coaxial cable is a common medium used for computer and cable television networks. It is better shielded than the twisted pair and allows for higher transmission speeds (up to one or two gigabits per second).

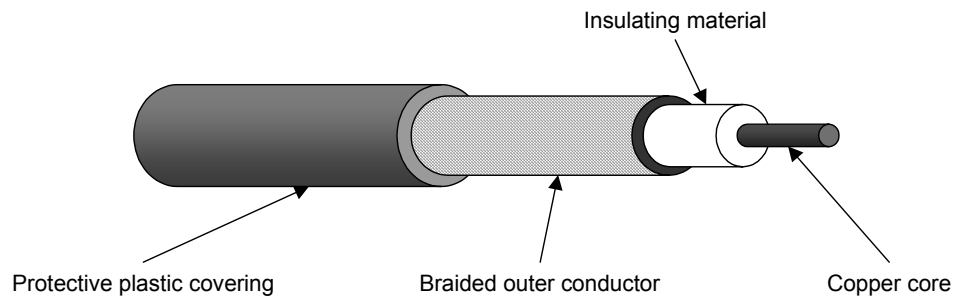


Figure 26: Coaxial cable

These cables consist of a stiff copper wire surrounded by insulating material. This is surrounded by a cylindrical conductor in the form of a braided mesh, which in turn is covered by a protective plastic sheath (Figure 26). Coaxial cables are distinguished into baseband coaxial cables for digital transmission and broadband coaxial cables for analogue cable television.

6.4.1.3 Fiber optics

Fiber cables are an optical transmission system. Such a system consists of three components: a light source, the transmission medium, and a detector. The light source is either an LED (light emitting diode) or a semiconductor laser. The transmission medium is an ultra-thin fiber of glass; the detector is a photodiode, which generates an electrical pulse when hit by light. Due to total reflection, light emitted at a source can travel within the fiber for many kilometers with virtually no loss. A fiber cable is similar to a coaxial cable. It has a glass core about the diameter of a human hair, surrounded by a glass cladding which serves to keep the light in the core. An outer plastic jacket covers the cladding and the core (Figure 27). As with twisted pairs, optical fibre cables are usually bundled together, protected by an outer sheath.

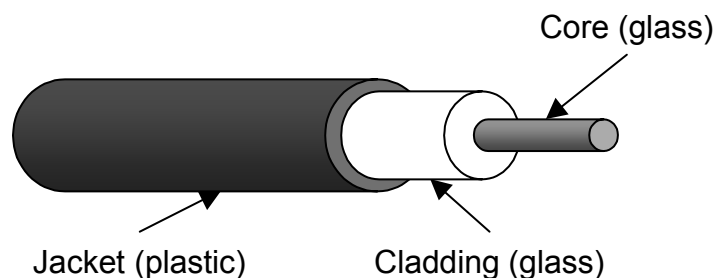


Figure 27: Fiber cable

With current technology, data transmission through fiber cables reaches a practical bandwidth of one gigabit per second, although theoretically a much higher transmission rate is possible. Fiber optics is used in local as well as long-haul transmissions. Apart from providing a larger bandwidth, fiber cables have other advantages over copper wires. They provide data transmission over longer distances without repeaters, they cannot be affected by power surges or electromagnetic interference, they are thin, and do not leak light. Moreover, fiber cables are more difficult to tap than copper wires.

6.4.2 Wireless transmission

Electromagnetic waves are used for wireless transmissions. Their advantage is that once broadcast by a transmitting antenna, they travel with the speed of light and can be received by a receiver some distance away. Depending on the range used within the electromagnetic spectrum, we speak of radiowaves, microwaves, infrared-, visible light, ultraviolet light, X-rays, and gamma rays. Only radio- and microwaves as well as infrared and visible light are in fact used for transmitting information. The other parts of the spectrum are dangerous to living things.

The advantage of wireless media is that cables do not have to be installed and one transmitter can, in principle, reach many receivers. Depending on the wavelength used, a repeater must be installed at certain distances. For the frequently used microwave links, a repeater is required about every 50 km. This medium is used mainly for data and voice transmission in the telecommunication business. Infrared transmission is well known for its use in remote control of video and audio equipment within a room. It is also used for linking printers to notebook computers. The nature of infrared requires, however, that the transmitter is in sight of the receiver and that the distance between them is not too large. In the visible light range, lasers are used as a transmission medium. Communication satellites receive a microwave signal, amplify it and rebroadcast it back to the surface of the Earth.

Wireless transmission was a booming market in the 1990s and became even more popular today for wireless local area networks. Notebook and hand-held computers are linked through modems to the systems in the office or base by using telephone lines. These lines can be analogue or digital. A bandwidth of 2 Mbps is typically achievable.

A modem connection via a standard analogue telephone line can reach 28.8 kbps, and with compression and error correction features built in, a slightly higher effective data rate can be achieved. An alternative to the analogue telephone network is the Integrated Services Digital Network (ISDN). Its goal is to provide voice and data transfer simultaneously on a digital network. ISDN comes in two versions, narrowband ISDN (N-ISDN) and broadband ISDN (B-ISDN). The former provides only 64 kbps, whereas the latter is designed for a bandwidth of 155 Mbps. N-ISDN is virtually useless for modern high speed networks, whereas B-ISDN based on Asynchronous Transfer Mode (ATM) technology meets their requirements. The slower N-ISDN can, nevertheless, be very useful for the home user who wishes to download web pages at a higher speed than possible with a 28.8 kbps analogue modem. In recent years, Asynchronous Digital Subscriber Lines (ADSL) are beginning to replace ISDN connections for Internet access. ADSL offers a download speed of 768 kbps and an upload speed of 128 kbps.

6.4.3 Network hardware

We can classify computer networks according to the transmission technology they use, the scale at which they operate, and the network topology, i.e., the shape of the graph of pairwise connections between computers.

There are two types of transmission technology: (1) broadcast networks, and (2) point-to-point networks. In broadcast networks short messages, called packets, are sent from one computer to all other computers. These packets contain address information identifying the computer for which they are intended. Only this computer will process the packets, all others will ignore them. This technique is called broadcasting. If a subset of computers in the network is addressed, this is called multicasting. In a point-to-point network, each computer is linked to one or more computers and communication is between pairs of machines. If, for instance, a message is to be sent from computer A to computer B, it may travel through intermediate machines before it reaches the destination computer. Routing algorithms searching the optimal path through the network are of particular importance. Larger networks are mainly point-to-point, whereas smaller, geographically limited networks use broadcasting.

Another way of classifying networks is according to their scale. Networks of up to a size of 1 km are known as Local Area Networks (LANs), while Metropolitan Area Networks (MANs) roughly span the area of a city (up to 10 km). Larger areas such as countries or continents are networked with Wide Area Networks (WANs), and at a global scale they are called internetworks. The Internet is an obvious example of the latter.

The structure of nodes and links between them is called the network topology. Figure 28 shows the most common network topologies: star, ring, tree, complete graph, and bus.

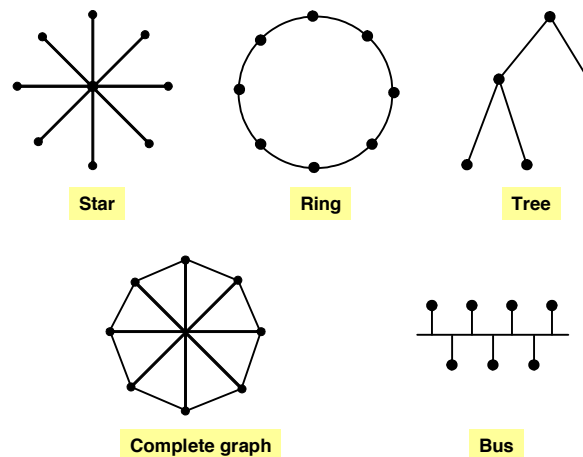


Figure 28: Network topologies

6.4.3.1 Local area networks (LANs)

Local area networks usually link computers within a building or on a campus. They are mainly used to connect PCs, workstations, and other devices such as printers. LANs are broadcast networks, using a single cable to which all computers and devices are linked (bus or ring topology). In a bus topology, one machine is allowed to send at a time, the others have to wait. If two or more machines want to transmit

simultaneously, an arbitration mechanism has to resolve the conflict. This mechanism can be centralized or distributed. A typical example is the standard IEEE 802.3 or Ethernet. It uses a decentralized arbitration mechanism and operates at a bandwidth of 10 to 100 Mbps. Another type of topology for a LAN is the ring. The IEEE 802.5 (or IBM token ring) operates at bandwidths of 4 and 16 Mbps.

6.4.3.2 Metropolitan area networks (MANs)

Many commercial activities are expected to take place in metropolitan areas making use of computer network technology. Increasingly, cable companies open their networks for other services such as Internet access, telecommunications, and emergency services. As in LANs, the transmission mode in MANs is broadcast or multicast. For MANs an international topology standard has been adopted as IEEE 802.6 under the name of Distributed Queue Dual Bus (or DBDQ), consisting of two buses with unidirectional communication.

6.4.3.3 Wide area networks (WANs)

Wide area networks span a large geographical area and contain a collection of machines called hosts or end systems. The hosts usually belong to a LAN in which there is a special computer, called a router, that connects the LAN to a communication subnet which, in turn, connects to other LANs (Figure 29). The routers are also called packet switching nodes. Routers that are not directly connected, communicate indirectly via other routers through a store-and-forward mechanism. The subnets are typically point-to-point networks. Common topologies for WANs are star, ring, tree, or complete graph. Irregular topologies, often reflecting the historic growth of the network, are also common.

Another possibility for WANs are satellite systems. These are broadcast systems in which many ground stations can listen to the satellite transmission, and they are usually part of a bigger network.

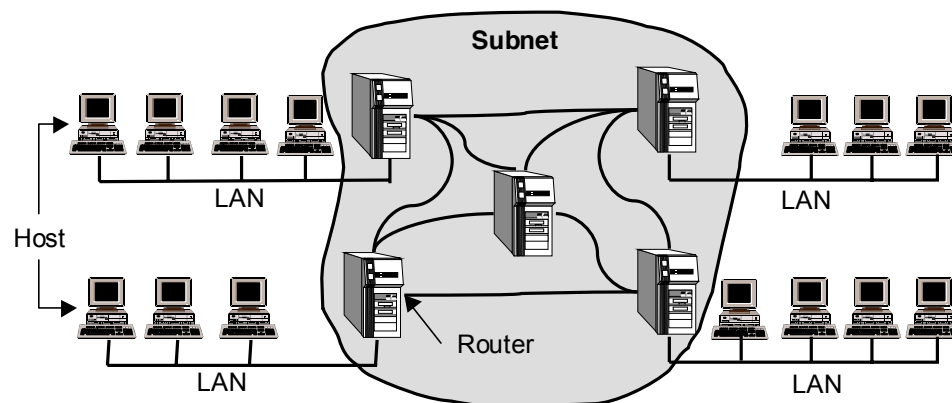


Figure 29: Wide area network

6.4.3.4 Internetworks

Several heterogeneous networks connected through gateways are called an internetwork. In analogy to WANs, where a router connects LANs, in an internetwork the gateways connect WANs. The Internet is perhaps the best known, but not the only example. It is a global internetwork whose origin dates back to a project by the American Department of Defense when the Defense Advanced Research Projects Agency (or DARPA) launched the ARPANET in 1969. By the early 1980s, the ARPANET had grown steadily to include hundreds of host

computers and several of today's key protocols were stable and had already been adopted.

When the supercomputer network of the American National Science Foundation NSFNET was connected to the ARPANET the growth became exponential. Many other networks joined such as BITNET, HEPNET, and EARN. By 1990, the ARPANET was shut down and dismantled. Nowadays there are millions of computers connected to the Internet and high-speed links connect continents running from 64 kbps to 2 Mbps.

The Internet has many applications that are indispensable in today's information infrastructures: electronic mail, news, remote login, file transfer, and the World Wide Web. The application with the strongest impact on society is the WWW. Since its invention in the early 1990s and the introduction of browsers (such as Mosaic, Netscape Navigator, and Internet Explorer) websites have been created at an enormous rate. Almost every company, organization, government, and university, but also private individuals, have their own home page. And their number is increasing.

A computer is said to be linked to the Internet when it has an IP address, runs the TCP/IP protocol stack, and can send IP packets to all other machines on the Internet. The next section provides a more detailed explanation of the terms used here.

6.4.4 Network software

When computers communicate with each other they must 'speak' the same language, and implement correct procedures to establish and maintain a connection. Computer networks and network software are complex. They are therefore organized as a series of layers or levels. Each layer offers certain services to higher layers. Interfaces between layers define which primitive operations a layer offers to a higher layer. Layer n on one computer (host) communicates with layer n on another computer according to certain conventions and rules. These rules are called protocol. The respective layers communicating with each other are called peers. The peers do not communicate directly. Information is passed to lower layers through the interfaces to the physical medium (below layer 1) where the actual communication takes place. Figure 30 illustrates this concept.

A network architecture is a set of layers and protocols. Details of implementation and specifications for interfaces are not part of a network architecture. They may actually vary among systems. A set of protocols for a certain system, one for each level, is called a protocol stack.

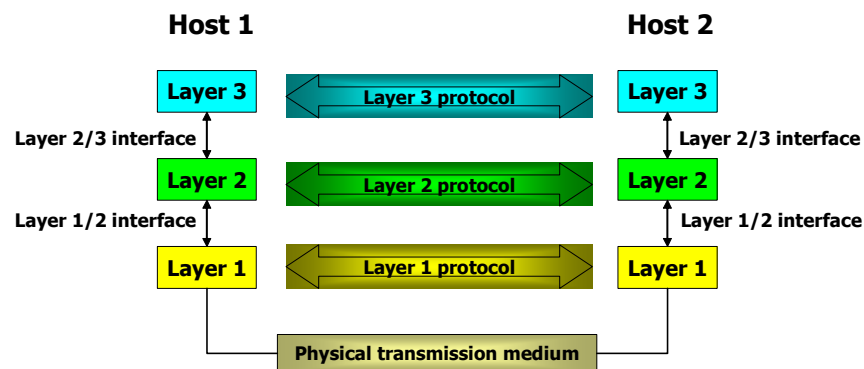


Figure 30: Layers, protocols and interfaces

6.4.4.1 Layer Services

Communication between computers means sending messages from one machine to another. If travel in only one direction is allowed we speak of simplex communication. A transfer in both directions, but not simultaneously, is called half-duplex communication, and if there are no restrictions, we have full-duplex communication.

With so many computers involved, it is extremely important to have an addressing mechanism that allows unique identification of senders and receivers. Rules must be defined for how data can travel between layers. Other services refer to error-detection and error-correction, disassembling and reassembling long messages, avoiding data overflow due to fast transmitters and slow receivers, and routing of messages.

There are two models of significance for modern computer networks: the ISO/OSI and the TCP/IP reference models. The first is an international standard, yet of little practical importance; the latter is less of a model but it is widely used in the Internet. The next section discusses both models in detail.

6.4.4.2 The OSI Reference Model

The Open Systems Interconnection (OSI) reference model is a standard approved by the International Standards Organization (ISO). It was developed in the early 1980s and comprises two major components: an abstract model of networking (the seven-layer model) and a set of protocols. The model divides a networking system into layers. Every layer has a certain functionality and communicates with the layer directly under it through an interface. Layers communicate with their corresponding peers in other systems. The ISO/OSI model has a stack of seven layers on top of the physical medium (Figure 31): physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and application layer.

This model is important in that it is a means to explain and describe network architectures. Its protocols are of minor importance. They have been overtaken by the TCP/IP protocols that play an important role in the Internet.

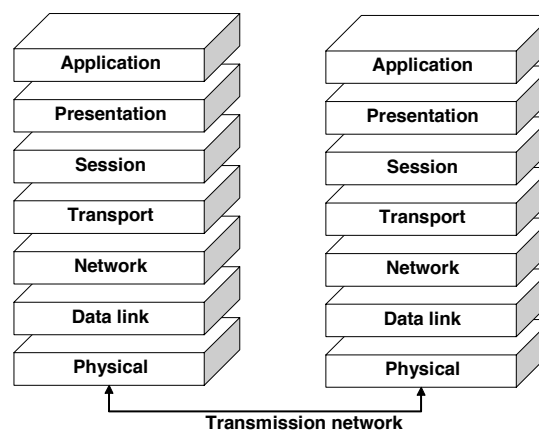


Figure 31: ISO/OSI reference model

Physical layer. The physical layer is mainly concerned with the mechanical and electrical interfaces on top of the physical medium that guarantee that bits are transmitted correctly over a communication channel. For instance, this layer defines the type of cable connectors, size of cable, or the voltage and duration to represent 0 and 1 bits.

Data link layer. The data link layer describes the logical organization of data bits that are transmitted through a channel such that an error-free transfer is achieved. This is done by the sender breaking the bit stream into data frames (up to 10 kByte in length) and process acknowledgement frames that are sent back by the receiver. Correction of damaged, duplicate, or lost frames is another task of the data link layer. It must also exercise buffer management, which means alleviating transmission speed differences between sender and receiver.

Network layer. Whereas the data link layer is only responsible for moving frames from one end of a cable to the other, the network link layer must take care of the proper operation of the communication subnet by routing packets from the source to the destination. Routing can become an important issue when accounting and billing are involved. Another important task is to resolve naming conflicts (addressing schemes may be different among heterogeneous networks) and differences in protocols.

Transport layer. The transport layer describes the nature and quality of the data delivery. Its major task is to accept data from the session layer (above it), split it into pieces, pass them through to the lower layers, and ensure the correct arrival at the remote end. This layer is the first with a true end-to-end communication, i.e., the transport layer on the source machine conceptually communicates with the transport layer on the destination machine, without a concept of intermediate nodes. Another function of the transport layer is flow control, to regulate the flow of information, and to establish and delete connections across the network.

Session layer. The session layer's prime task is to accommodate sessions between users on different computers. Sessions might be ordinary data transport, remote login, or dialogue control. Other functions are token management and synchronization. The session layer provides tokens that regulate who is allowed to perform certain actions. Only the holder of a token is allowed to perform a critical operation, thereby avoiding that two machines attempt such an operation at the same time. Synchronization establishes checkpoints that help in the recovery of network failures. When a large data set is being transferred from one machine to the other, it is sometimes interrupted by network failures. If this happens, only the data that were lost after the last checkpoint have to be re-transmitted.

Presentation layer. The presentation layer provides support for some common functions that have to know the syntax and semantics of the data. Character encoding for text (such as ASCII), dates, or floating point numbers are done here.

Application layer. The application layer is the level that real users see when they work in a computer network. Several application protocols are supported that directly target the end user. These protocols support file transfer between different operating systems and take care of problems that may arise due to incompatible file naming conventions (imagine, for instance, the different file naming conventions in DOS and UNIX). Other functions supported are electronic mail, remote job entry, or directory lookup. It must be noted, however, that most of the protocols we use on the Internet are not OSI protocols, but TCP/IP model protocols, which are explained in the next section.

6.4.4.3 The TCP/IP Reference Model

When the Internet evolved from the old ARPANET, and other networks were added, it soon became necessary to design a framework to link these heterogeneous networks. The TCP/IP reference model, named after its two major protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP), was defined in 1974. Figure 32 shows a comparison between the ISO/OSI and the TCP/IP reference model layers.

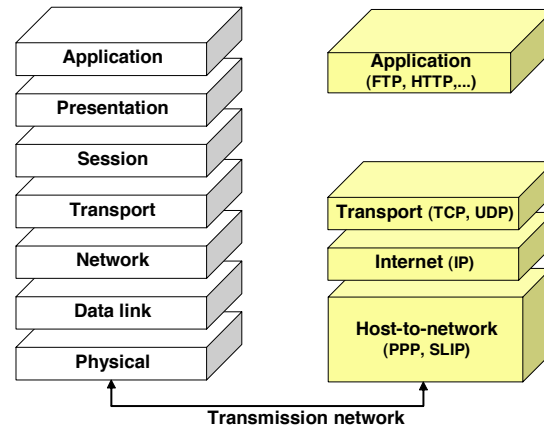


Figure 32: ISO/OSI and TCP/IP layers

We see that the TCP/IP reference model has fewer layers than the ISO/OSI model. In particular, the network access layer (host-to-network) is vaguely defined. The model only requires some protocols to take care of a correct connection to the network and to guarantee that the data packets from the Internet layer are transmitted correctly. One of these protocols that is widely used to connect home computers through telephone lines to host computers of Internet providers is the Point-to-Point Protocol (PPP).

Internet layer. The Internet layer supports the Internet Protocol (IP) as the most basic means of transportation of data packets. IP is a protocol that treats each packet independently. It does not even check whether the packets reach their destination and takes no corrective action if they do not arrive. IP is basically an unreliable protocol because it contains no error detection or correction. It provides the following services:

Addressing. The sending and receiving hosts are identified by 32-bit IP addresses. These addresses are used by intermediate routers to channel the packets through the network. Domain names are alphanumeric strings identifying Internet hosts. Examples of domain names are `www.itc.nl` or `hp18.itc.nl`. In this example, `itc` is a subdomain of the top-level domain `nl`. There are two types of top-level domains: generic and countries. Generic domains are `com` (commercial), `edu` (educational institutions), `gov` (US government), `org` (non-profit organizations), `int` (international organizations), `mil` (US armed forces), and `net` (network providers). Country domains use the ISO 3166 country codes. Generic domains are generally used in the United States, country domains elsewhere. Domain names are converted into IP addresses, a 32-bit number used to identify an Internet host through the Domain Name Service (DNS). IP addresses are written in a 'dotted quad' notation, e.g., `127.18.53.10`.

Fragmentation. The IP splits data packets into small pieces, thus allowing larger packets to travel on networks that can only handle smaller packets. Fragmentation and defragmentation is done transparently.

Packet timeouts. Every IP packet contains a counter that is decreased whenever it passes a router. This prohibits packets travelling in circles forever.

Type of service. Every IP packet can indicate that it requires a particular type of service, such as a high transmission speed more than reliability (for voice transfer) or high reliability more than speed (for file transfer).

Transport layer. The transport layer allows two corresponding peers in a network to communicate. As in the ISO/OSI model, end-to-end protocols have been defined for the TCP/IP model. The two protocols used are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

The TCP is a reliable connection-oriented protocol that allows two peer entities to communicate. It makes up for the deficiencies of the IP. The UDP, however, is used for the types of communication where speed is of higher priority than accurate delivery. This protocol is therefore often used for transmitting speech or video.

Application layer. There are no session and presentation layers in the TCP/IP model. In the application layer we find user-level protocols that allow remote login, file transfer, electronic mail, and more. The following is a list of the most important of these protocols:

TELNET. This is a simple terminal-oriented remote login service. It was and still is popular with text-oriented terminal emulations on remote machines.

FTP. The File Transfer Protocol is one of the oldest Internet protocols and is still in widespread use. It is based on the TCP and is used for the transfer of text and binary data files. Normally, users must identify themselves with a username and password on the host machine when they want to transfer data. Anonymous FTP is a variant of FTP that allows users to access host machines without having a personal account on that particular computer.

SMTP is the Simple Mail Transfer Protocol that allows the sending of electronic mail.

DNS is the Domain Name Service that maps domain names to their corresponding IP addresses.

NFS, the Network File System, is a service that allows TCP/IP clients to share files among each other.

NNTP is the Network News Transfer Protocol that is used to handle newsgroups. Electronic mail normally transfers messages from one sender to one or more receivers. News messages could potentially reach every Internet host and user.

HTTP (or Hypertext Transfer Protocol) is the youngest of the Internet protocols. It allows communication of multimedia and hypermedia documents that are made available mainly through the services of the WWW. These documents are mostly written in HTML (Hypertext Markup Language).

6.4.4.4 Comparison between the OSI and TCP/IP Reference Models

Both reference models have much in common: the concept of a stack of independent protocols and similar layer functionality. TCP/IP, however, is older and was designed with an already operational network (ARPANET) in mind. The ISO/OSI model is younger and was developed as a reference model with the aim of defining (not necessarily building) a network architecture. The ISO/OSI model is therefore a better vehicle to explain and describe network architectures, although it is of less practical importance. The TCP/IP model reflects a pragmatic approach and can teach us a lot about practical network design.

The OSI model makes a clear conceptual distinction between services, interfaces and protocols. The protocols of the TCP/IP model are not well hidden and they are less clear in distinguishing different concepts.

6.5 The World Wide Web

One of the recent Internet developments which has had an enormous impact on the market as well as on society is the World Wide Web (WWW), or as it is generally called, the Web. It was started at the European centre for nuclear research, CERN, out of the need to exchange documents of various kinds among scientists spread over many countries. A first text-based prototype was demonstrated in late 1991, and the first graphical interface, called Mosaic, was released in 1993. Such interfaces became known as web browsers. Today, two browsers dominate the market, Netscape Communicator from Netscape Communications Corporation, and Microsoft's Internet Explorer. The World Wide Web Consortium was established in 1994 to further the development of the Web. Its home page can be found at <http://www.w3.org>.

The Web is a client-server system; on the client side it presents itself to the user as a vast collection of web pages. These pages contain hyperlinks to other pages. Besides text, different media may be used to present the contents of a page, such as sound, video, or images. We then speak of hypermedia. To view different types of media, the browsers need external viewers, or helper applications. Frequently these helper applications are built into the browser as a plug-in.

Machines accessing the Web through a browser must be directly connected to the Internet, or at least able to establish a PPP connection with an Internet provider. This is necessary because browsers establish TCP connections with Web servers to access pages. Sites that provide pages on the Web run a server process that listens to requests from clients (browsers). The hypertext transfer protocol (see above) handles these requests.

Web pages are addressed through a Uniform Resource Locator (URL). For instance, the home page of the University of Vienna has the URL <http://www.univie.ac.at/home.htm>. A URL consists of three parts: the name of the protocol (http), the host name where the page resides (www.itc.nl), and the name of the hypertext document (home.htm). The following is what happens when a web page is displayed in a browser window:

1. A user clicks a hyperlink or enters a URL.
2. The browser determines the URL (when a hyperlink was clicked).

3. The browser asks the domain name service (DNS) for the IP address of the host computer.
4. DNS resolves the name and returns the IP address.
5. The browser establishes a TCP connection to the host.
6. The browser requests the document.
7. The host server sends the requested file.
8. The TCP connection is closed.
9. The browser displays all the text of the document.
10. The browser fetches and displays all images in the document.

Web pages are written in HTML (HyperText Markup Language) or Java. HTML is useful for writing static web pages. Java, developed by Sun Microsystems, is an object-oriented programming language that allows a provider to write applets. A hyperlink can point to an applet that is downloaded and run by the browser locally. Applets provide interactivity and flexibility over static HTML pages. However, by executing 'foreign' code on a local machine, they also pose a potential security threat.

Organizational Infrastructure

GIS need an infrastructure that enables them to function. Whereas the technical infrastructure mainly deals with the hardware and network architectures. The organizational infrastructure addresses issues of procedures and policies to make GIS work.

The need to exchange data among different systems leads to the definition of standards for the transfer of spatial data. The purpose of these *transfer standards* is to move the contents of one GIS database to a different GIS database with a minimal loss of structure and information. Since the early 1980's, many efforts have been made to develop such standards on a local, national and international level. Today, we have operational transfer standards that support the dissemination of spatial data.

In a transfer process data is always physically moved from one system to the other. A completely different approach to data sharing is *interoperability* of GIS. Here, GIS software accesses data on different systems (connected through a computer network) through standardized interfaces. Data does not need to be physically transferred and converted. The Open GIS Consortium is the leading organization.

7.1 Database Technology

Geo-information infrastructures are based on large amounts of spatial data distributed over many agencies. It is important to provide an organization that supports reliable and efficient sharing of the data among many users. A simple collection of data files or a file directory is not sufficient for this purpose. The solution is database systems, one of the enabling technologies for geo-information infrastructures. A database system comprises a database and a database management system. The database (DB) is an organized collection of stored data, and a database management system (DBMS) is software for building and maintaining a database.

The advantages of using a database system are that the DBMS provides the following features: persistency, efficient storage management, recovery, concurrency control, ad hoc queries, and data security. A persistent storage means that data exist independently of the execution of a program, unlike data in program variables that do not survive the execution of the program. Transactions are operations that move a database from one consistent state to another consistent state. If, for whatever reason, a transaction cannot be successfully completed, the database is returned to the consistent state before the transaction (recovery). To avoid inconsistencies caused by concurrent read and write operations to the database by many users accessing the database at the same time a concurrency control mechanism is employed by the DBMS. This gives every user the impression that he/she is the only one accessing the database.

Not all queries can be known beforehand. A DBMS must provide means for formulating and satisfying ad hoc queries without having to write application programs. Query languages provide this kind of function. Furthermore, a DBMS must allow for multiple users with different rights for accessing the database (data security).

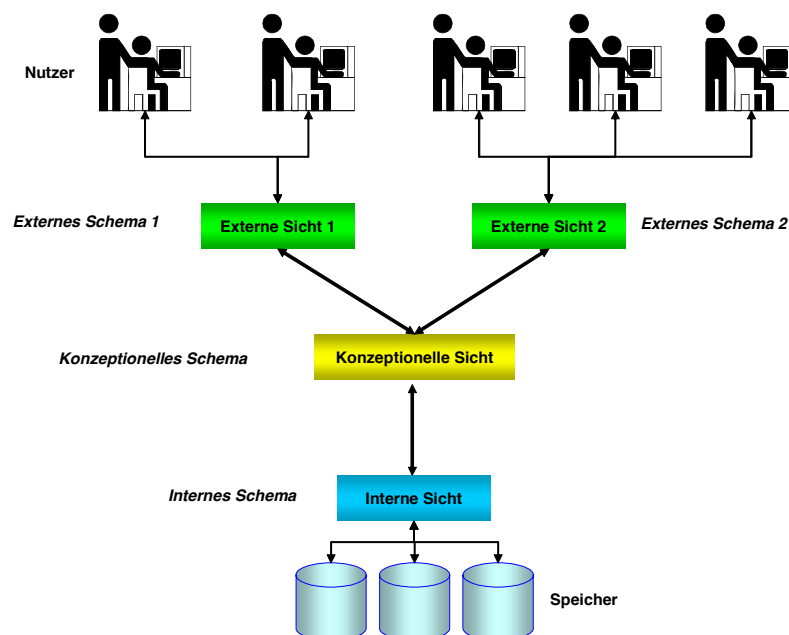


Figure 33: Database architecture

7.1.1 Database architectures

A database architecture is divided into three levels: internal, conceptual, and external level (Figure 33). This architecture was proposed by the ANSI/SPARC Study Group on Data Base Management Systems in 1978, and has been widely adopted. The external level is closest to the users; the internal level is closest to the physical storage. Individual users may have different external views on the database. There is only one conceptual view and one internal view. The views are defined by schemas using a data definition language (DDL). The data manipulation language (DML) is used to describe database objects' processing.

7.1.2 Data models

Databases store models of a perceived reality. In the design process of a database, several data models are being used to describe subsets of the real world at different levels of abstraction. A data model describes the contents, structure, and meaning of data. They can be classified into conceptual-, logical-, and physical data models (Figure 34). A database designer starts with an abstract subset of the real world (external model or application model) which is subsequently mapped into a conceptual model. The best known conceptual data model is the entity-relationship model. This, in turn, is mapped into a logical data model, which can be implemented with a commercial database management system. The most popular is the relational data model, which was introduced in 1970. It uses a simple data structure, the relation. Relations are tables, a concept that is easily understood by human common sense.

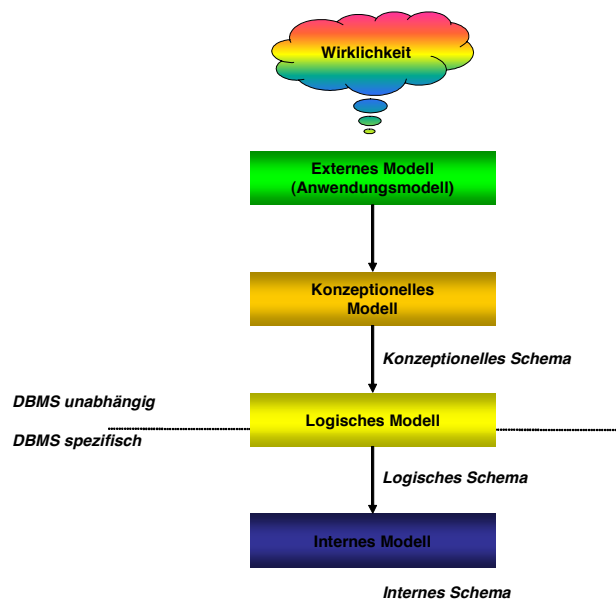


Figure 34: Models in database design

The database language that supports the relational data model is SQL (or Structured Query Language). It became an ANSI standard as SQL1 in 1986 (ANSI X3.135) and an ISO standard in 1987. A revised and extended version of the standard was approved as SQL2 (also known as SQL92) in 1992, and SQL3 is nearing its completion. SQL is both a data definition and data manipulation language and is supported by all major relational DBMS vendors.

7.1.3 Distributed databases

In a geo-information infrastructure, many databases are distributed over numerous organizations and agencies. A distributed database consists of a number of sites interconnected through a communication network. Every site runs an autonomous database management system. Local applications run only on a local database, global applications span some or all the sites of the distributed database.

A distributed database looks like a local database to the user. Important concepts of distributed databases are distribution transparency, fragmentation, and replication. Distribution transparency means that a user does not see where the data in the database are physically located. Even if the same data are stored in duplicate at different locations for reasons of efficiency (replication) or different columns of a table are physically located at different local databases (fragmentation), the user always perceives a central database.

Whereas traditional database design involves conceptual-, logical-, and physical design, for distributed databases we also have distribution design. There are two possible approaches to the design of distributed databases: top down and bottom up. The top-down design is applied to new databases that are designed from scratch (Figure 35).

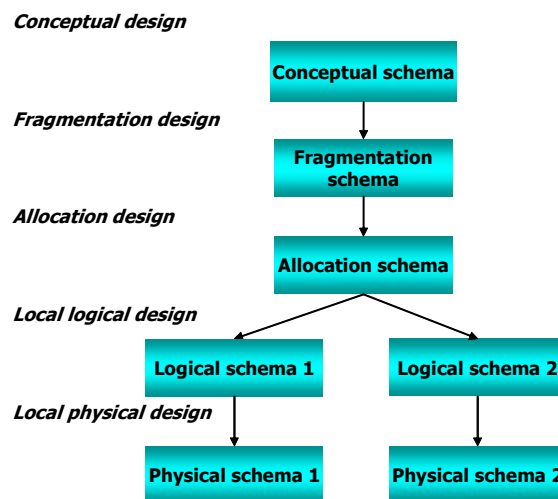


Figure 35: Top-down distributed database design

Very often, however, individual databases already exist, and have been used as such for a long time. When the need arises to integrate these databases for a higher level application, a special kind of distribution architecture comes into place which is called federated databases. The federated database design approach is presented in Figure 36.

7.2 Clearinghouse

The United States Federal Geographic Data Committee (FGDC) defines a clearinghouse as a 'system of software and institutions to facilitate the discovery, evaluation, and downloading of digital geospatial data.' Such a clearinghouse usually consists of a number of servers on the Internet that contain information about available digital spatial data known as metadata.

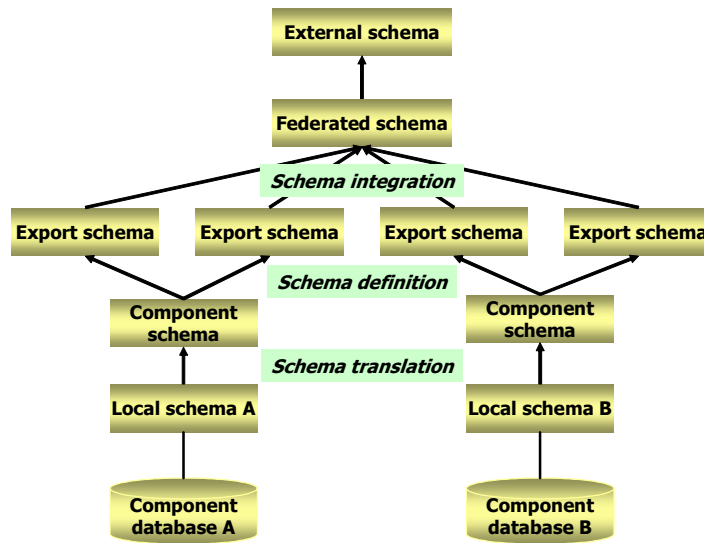


Figure 36: Federated database design

A clearinghouse is an example of a client-server architecture. The server machines hold the metadata, and the clients request information about the availability of spatial data by visiting the server nodes, usually through a Web browser. In the ideal case not only the metadata but also a link to the data itself is provided by the server. There is no hierarchical relationship among the servers. They act as peers within the clearinghouse activities. Servers are usually installed at providers, such as national mapping agencies, at different organizational levels. A typical clearinghouse architecture consists of many interoperating metadata servers.

Providers of a clearinghouse must have access to a machine directly connected to the Internet. This server holds the metadata and must run server software. One protocol selected to provide search interoperability among different servers is the ISO 10163-1995 (or ANSI Z39.50-1995) search and retrieve protocol. The Z39.50 was initially developed for the library community. It contains client and server software that establishes a connection, relays a query, returns the query result, and presents retrieved documents in various formats. Figure 37 shows the clearinghouse concept.

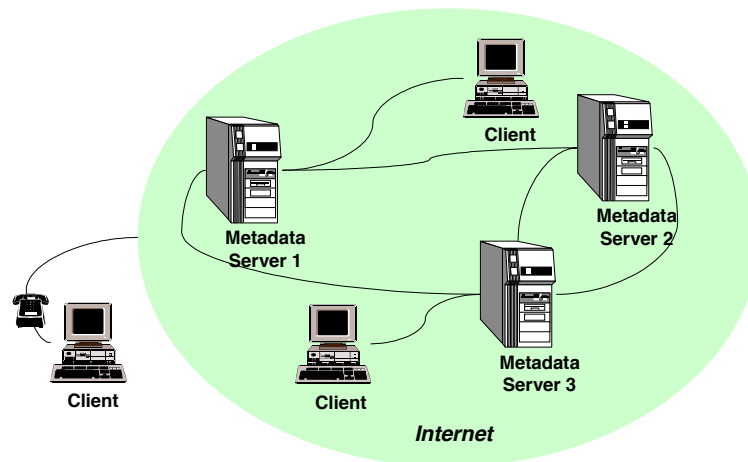


Figure 37: Clearinghouse concept

7.3 Spatial Data Transfer

When we transfer data from one system to another, we might encounter various problems: transfer media are not compatible, physical and logical file formats are different, we might not have any information concerning the quality of the dataset. Moreover, do we need translators from and into every format that we might have to deal with. In the worst case, we need for n different systems $\frac{n(n-1)}{2}$ translators. Using a standard reduces this number to $2n$, because we only need one translator—between the system and the standard—for each system¹². Figure 38 illustrates this situation.

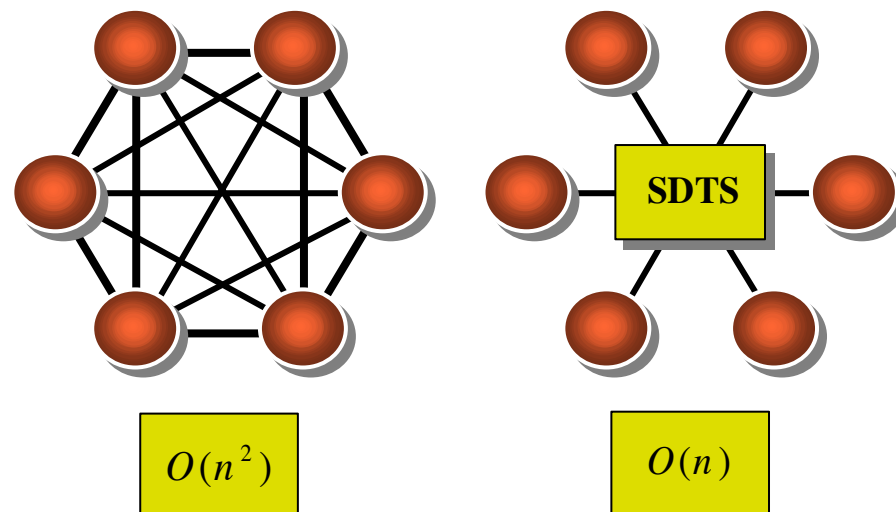


Figure 38: No standard versus a transfer standard

The solution to these problems is to exchange spatial data in a standardized way thereby keeping as much as possible of the structure and relationships among the features in the dataset.

7.3.1 Spatial Data Transfer Process

A data transfer standard defines a *data model* for the transfer as well as a *translation mechanism* from a GIS to the *exchange format* and into the exchange mechanism. In transferring data from system X to system Y we need a translator that converts the contents of the database in system X into the model of the spatial data transfer standard (SDTS)¹³. The components of the model are represented by modules that are converted into a computer readable format. A standard often used is the ISO 8211 Data Descriptive File. On the receiving side, system Y needs a translator that converts the data from the transfer standard into the database. In the ideal case, after the transfer is completed, no further processing is needed. The data are ready to use.

¹² We need n translators to translate into the standard and n to translate from the standard, which makes it $2n$.

¹³ We use here SDTS as abbreviation for the term spatial data transfer standard. This should not be confused with the FIPS 173 SDTS, the federal spatial data transfer standard of the United States of America.

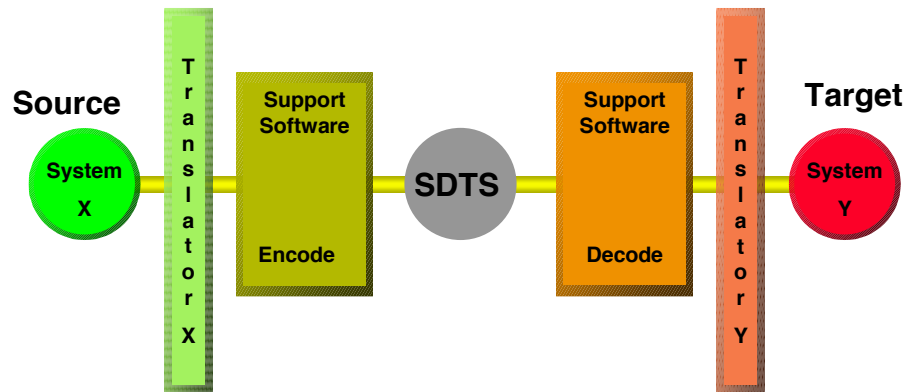


Figure 39 Spatial data transfer process

7.3.2 Examples of Spatial Data Transfer Standards

The following table gives an overview of some exchange standards of different countries and organizations:

Table 11: Examples of spatial data transfer standards

NATO	DIGEST (<u>D</u> igital <u>G</u> eographic <u>E</u> xchange <u>S</u> tandard) of the <u>D</u> igital <u>G</u> eographic <u>I</u> nformation <u>W</u> orking <u>G</u> roup (DGIWG); members are Belgium, Canada, Denmark, France, Germany, Italy, The Netherlands, Norway, Spain, United Kingdom, USA
United Kingdom	NTF (National Transfer Format)
USA	SDTS (Spatial Data Transfer Standard)
France	EDIGéO

These standards are accepted as federal or national standards based on international ISO standards.

Industry standards are for instance DLG (digital line graph of the USGS), DXF (AutoCAD Format). Industry standards are frequently used standards that were introduced by a company or organization but which are not accepted as national or federal standards.

GIS Technology in Organizations

Spatial information infrastructures (also called geoinformation infrastructures) are being built in many countries. They comprise the setting up of an organizational and technological framework to share and disseminate geo-spatial data. One crucial part in this endeavor is the introduction of GIS technology into an organization.

This chapter describes the problems encountered and procedures to be followed when an organization goes GIS.

8.1 Prerequisites and Problems

In setting up spatial information infrastructures issues of organization, policy, hardware and software procurement, networking, training and education have to be addressed. The implementation of GIS software is one of the important steps in this process.

The implementation of a GIS requires a thorough analysis of the structure, workflow and equipment needed of the organization where the system will be installed. Experience shows that certain problems occur frequently in implementing a GIS. The following points should always be considered:

A GIS is not only hard- and software, but requires also personnel and organization.

Many customers have big problems in describing the functioning, workflow, and required products of their own organization.

Hard- and software is often procured without a previous analysis and only then, it is considered what could be done with the system.

Very often, it is overlooked that a system without data is useless. Data capture is up to ten times more expensive than the equipment.

An “objective” selection of systems is very often impeded by policy decisions.

Trained personnel are very important for running the system. There are many different tasks, e.g., a digitizing operator needs not know anything about databases, but has to know the digitizing sub-system of the GIS very well.

8.2 Implementation Strategy

The successful implementation of a GIS can be described as a six-step process (after ARONOFF 1989).

Awareness. The advantages of GIS are recognized and potential users are identified.

Developing system requirements. A project for the introduction of GIS is defined and systematic work is started, often by installing a “modernization committee” (information gathering, user requirements, visit to trade fairs, product demonstrations).

System evaluation. Based on the system requirements potential systems are selected and subjected to a *benchmark test*. Sometimes it is also useful to conduct a pilot study.

System justification and development of an implementation plan. After system evaluation a decision has to be made in favor or against the use of GIS technology. This can be supported by a cost-

benefit analysis. If a decision is made for GIS then an implementation plan must be designed.

System acquisition and start-up. System acquisition and installation, training, building of the database

Operational system. The database is ready, work procedures are automated and standardized, training and education of staff, maintenance

References and Bibliography

- Aronoff S., 1989, *Geographic Information Systems: A Management Perspective*. WDL Publications, Ottawa.
- Burrough P.A. 1986. *Principles of Geographical Information Systems for Land Resources Assessment*. Clarendon Press, Oxford.
- Burrough P.A., McDonnell R., 1998, *Principles of Geographical Information Systems*. 2nd edition. Oxford University Press, Oxford, ISBN: 0198233655
- Corbett, J.P. 1979. *Topological Principles in Cartography*. Technical paper – Bureau of the Census; 48.
- Dangermond, J. 1984. Review and Synthesis of Problems and Directions for Large Scale Geographic Information System Development. In: *Proceedings of the International Symposium on Spatial Data Handling*, Zurich, Vol. 1, pp. 19–25.
- Ehlers, M. and S. Amer. 1991. Geoinformatics: An Integrated Approach to Acquisition, Processing and Production of Geo-Data. *Proceedings, EGIS'91*, Brussels, Belgium, pp. 306–312.
- Goodchild, M.F. 1992. Geographical Information Science. *Int. J. Geographical Information Systems*, Vol. 6, No. 1, pp. 31–45.
- Groot, R. 1989. Meeting Educational Requirements in Geomatics. *ITC Journal*, No. 1, pp. 1–4.
- Guptill, S.C. and J.L. Morrison (eds.) 1995. *Elements of Spatial Data Quality*. Elsevier Science Ltd.
- Kainz W., Egenhofer M., Greasley I. 1993, Modeling Spatial Relations and Operations with Partially Ordered Sets. *International Journal of Geographical Information Systems*, Vol. 7, No. 3, 215–229.
- Kainz W. 1995, Logical Consistency. In: S.C. Guptill and J.L. Morrison (eds.), *Elements of Spatial Data Quality*, Elsevier Science.
- Langran G., 1992, *Time in Geographic Information Systems*. Taylor & Francis, London, Washington.
- Laurini R., Thompson D., 1992. *Fundamentals of Spatial Information Systems*. The A.P.I.C. Series Number 37, Academic Press.
- Longley P.A., Goodchild M.F., Maguire D.J., Rhind D.W., 2001, *Geographic Information Systems and Science*. John Wiley & Sons, Chichester.
- Nunes J., 1995, General Concepts of Space and Time. In: A. U. Frank (ed.), *Geographic Information Systems - Materials for a Post-Graduate Course*, Vol. 1: Spatial Information. Department of Geoinformation, Technical University of Vienna. Geo info 4.
- Pavlopoulos A., Theodoulis B., 1998, Review of SpatioTemporal Data Models. TimeLab Technical Report. Department of Computation, University of Manchester Institute of Science and Technology, 52pp.

- Peuquet D.J., Duan N., 1995, An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *Int. J. Geographical Information Systems*, Vol. 9, No. 1, 7-24.
- Peuquet D.J., Marble D.F. (eds.), 1990, *Introductory Readings in Geographic Information Systems*. Taylor & Francis.
- Samet, H., 1990a, *The Design and Analysis of Spatial Data Structures*. Addison-Wesley.
- Samet, H., 1990b, *Applications of Spatial Data Structures*. Addison-Wesley.
- Star J., Estes J., 1990, *Geographic Information Systems, An Introduction*. Prentice Hall.
- Tomlin C.D., 1990, *Geographic Information Systems and Cartographic Modeling*. Prentice Hall.
- Worboys M.F., 1992, A model for spatio-temporal information. In: B. Bresnahan, E. Corwin and D. Cowen (eds.), *Proceedings of the 5th International Symposium on Spatial Data Handling*. Humanities and Social Science Lab, University of South Carolina, Vol. 2., 602-611.
- Worboys M.F., 1994, Unifying the spatial and temporal components of geographical information. In: T.C. Waugh and R.G. Healey (eds.), *Advances in GIS Research, Proceedings of the 6th International Symposium on Spatial Data Handling*, Vol. 1., 505-517.
- Worboys M.F., 1995, *GIS: A Computing Perspective*. Taylor & Francis.

Index

- absolute space. *See* space
- ALU. *See* arithmetic and logic unit
- arc/node structure, 52
- arithmetic and logic unit, 62
- atom, 28
- atomist, 28
- attribute, 42
- Auto Carto, 15
- benchmark test, 92
- boundary, 38
- boundary model, 54, 55
- bucket methods, 55
- buffering, 60
- Burrough, Peter*, 16, 95
- cartographic generalization. *See* generalization
- cell, 38
- cell method, 48
- central processing unit, 62
- chain code, 53
- chronon, 43
- client, 67
- co-domain, 33
- communication devices, 63
- completeness. *See* data quality
- complex
 - cell, 38, 39
 - simplicial, 38
- computer architecture, 63
- computer network, 67
- computer science, 59
- consistency, 38
- contiguity, 60
- contour, 60
- Corbett, James*, 16, 95
- COSIT, 17
- CPU. *See* central processing unit
- Dangermond, Jack*, 16, 95
- Data, 27
- data handling, ix, 15, 21, 24, 57
- data model, 32
 - spatiotemporal, 42
 - topological, 54
- data quality, 23, *See* metadata
 - completeness, 23
 - lineage, 24
 - logical consistency, 23
 - positional accuracy, 23
 - purpose, 23
 - temporal accuracy, 23
 - thematic accuracy, 23
- data structures, ix, 47, 48, 52, 53, 55, 56
 - spatial, 16
- database, 17, 25, 48, 57, 58, 59, 83, 88, 93
 - relational, 16
- DIGEST, 89
- digital landscape model, 34
- digitizer, 65
 - cursor, 65
- digitizing, 18, 24, 52, 58, 92
- digitizing tablet. *See* digitizer
- DLG, 89
- DLM. *See* digital landscape model
- domain, 33
- DTM. *See* digital terrain model
- DXF, 89
- dynamic theory, 29
- edge matching, 58
- EDIGEO, 89
- elements, 29
- empirical reality, 30
- Euclidean geometry, 29
- Euclidean space, 37
- event-based model, 44
- feature, 22
- feature attribute, 22
- feature attribute type, 22
- feature relationship, 22
- feature type, 22
- Field-based models, 40
- film writer, 66
- floating-point unit, 62
- FPU. *See* floating point unit
- FTP, 67

- fuzzy logic, 35
- generalization, 32, 34, 59
- geographic information system, 12, 13, 18
- geographic space. *See* space
- geographic time. *See* time
- geoinformatics, ix, 13, 15, 17
- geo-information infrastructure. *See* spatial information infrastructure
- geo-information system. *See* geographic information system
- geomatics, 13
- geometric primitives, 17
- geo-relational model, 54
- GIS. *See* geographic information system
- granularity, 43
- graph, 39
 - planar, 40
- graph theory, 39
- grid directory, 51
- grid file, 51
- hardcopy, 19
- hardware, 62
- homeomorphism, 37
- HTML, 17
- HTTP, 17, 67
- identity of opposites, 28
- information, ix, 12, 17, 18, 19, 21, 22, 23, 52, 54, 57, 58, 83, 88, 91, 92
- information infrastructure. *See* infrastructure
- infrastructure, 15, 19
- interior, 38
- Internet, 17, 18
- interoperability, 83
- ISO 8211, 88
- Java, 17, 67
- k-d tree, 50, 51, 56
- keyboard, 64
- land information system. *See* geographic information system
- layers, 26, 58
- lineage. *See* data quality
- linear quadtree, 55
- linear scales, 51
- locations, 41
- logical consistency. *See* data quality
- map, 31
- mathematics, 15
- Maxwell, James Clerk*, 30
- memory, 62
- metadata, 22
 - application schema information, 23
 - data constraint information, 22
 - data quality, 22
 - distribution information, 22
 - extension information, 23
 - feature catalog information, 22
 - identification information, 22
 - maintenance information, 22
 - portrayal catalogue information, 22
 - reference system information, 22
 - spatial representation information, 22
- modem, 63
- monitor, 64
- morphism, 33
- Morton order, 15
- mouse, 64
- multi-purpose cadastre. *See* geographic information system
- NCGIA, 16
- network analysis, 60
- NTF, 89
- object, 42
- object-based models, 40
- octree, 55
- operating system, 66
- overlay, 59
- peripheral devices, 63
- phenomena, 26
 - man-made, 26
 - natural, 26
- Platonic solids, 29
- plotter
 - electrostatic plotter, 66
 - pen plotter, 66
- point quadtree, 49
- point-mode digitizing. *See* digitizer
- PR quadtree, 51
- printer
 - ink-jet printer, 65
 - laser printer, 65
- proximity, 60
- purpose. *See* data quality
- pyramid, 54
- quadtree, 15, 49, 50, 51, 53, 54, 55, 56
- real world, 22, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 58
- reasoning, 57
- region quadtree, 54
- register, 62
- relationship, 42
- relative space. *See* space, *See* space
- resolution, 64, 65
- Scaleless, 33
- scanner, 65

- schema
 - conceptual, 36
 - logical, 36
 - physical, 36
- SDH, 17
- SDSS. *See* spatial decision support system
- SDTS, 88, 89
- seamless, 33
- server, 67
- simplex, 37
- SIT. *See* spatial information theory
- skeleton, 39
- snapshot model, 44
- softcopy, 19
- software, 66
- space, 14, 27
 - absolute, 29
 - metric, 37
 - relative, 29
 - topological, 37
- space-time composite model, 44
- space-time cube model, 43
- spatial data. *See* data
- spatial decision support system, 15
- spatial information infrastructure, 92
- spatial information system. *See* geographic information system
- spatial information theory, 14, 16, 17
- spatial phenomena. *See* phenomena
- spatial reasoning, 14
- spatiotemporal data model. *See* data model
- spatiotemporal object model, 44
- SSD, 17
- storage devices, 63
- stream-mode digitizing. *See* digitizer
- SVGA, 66
- tabletop space. *See* space
- TCP/IP, 67
- temporal accuracy. *See* data quality
- TFT. *See* thin-film transistor
- thematic accuracy. *See* data quality
- theme. *See* layer
- thin-film transistor, 64
- time, 27
 - absolute, 43
 - branching, 43
 - continuous, 42
 - cyclic, 43
 - database, 43
 - discrete, 42
 - fixed, 43
 - implied, 43
 - linear, 43
 - relative, 43
 - transaction, 43
 - world, 43
- Tomlinson, Roger*, 15
- topological consistency. *See* consistency
- topological mapping. *See* homeomorphism
- topology, 15, 17, 35, 37, 52, 59
- transcendental idealism, 30
- UMTS. *See* Universal Mobile Telecommunications System
- uncertainty, 35
- Universal Mobile Telecommunications System, 68
- usage. *See* data quality
- VGA, 66
- view, 36
- visibility, 60
- void, 28
- voxel, 55
- WAP. *See* Wireless Application Protocol
- Wireless Application Protocol, 68
- workflow, 92
- World Wide Web, 17, 18
- WWW. *See* World Wide Web
- XML, 17
- XVGA, 66
- Zeno*, 28