Unit 051 - Information Organization and Data Structure

Written by: Albert K. Yeung
Ontario Ministry of Northern Development and Mines, Canada

Advanced Organizer

Topics covered in this unit

This unit presents an overview of the terminology and concepts pertaining to information organization and data structure in the context of information science and management. The aim is to provide a general but articulate introduction to the principles and methods of information organization, with special reference to geographic information, that serves as a prerequisite for more advanced studies of data models and database in subsequent units. Topics covered in this unit include:

- definitions of data, information, data files and database
- the concept and components of information domain
- the data-oriented approach to information system development
- the principles and methods of information organization
- the principles and methods of data structure
- design and specification of data structure by data modeling
- design and specification of data structure by process modeling

Learning Outcomes

After learning the material covered in this unit, students should be able to:

- distinguish between data and information
- describe the components of information domain in the context of information system

design
- describe the data-oriented approach to information system development
- explain information organization from the perspectives of: data, relationship, operating system and system architecture
- list typical spatial and non-spatial data structures and describe their characteristics
- describe the phases of work of data modeling and their respective end products
- describe the method of process modeling and its end product

Full Table of Contents

Metadata and Revision History

Unit 051 - Information Organization and Data Structure

1. Definitions and Terminology

1.1. Data and information

- many people use the terms "data" and "information" as synonyms but these two terms actually convey very distinct concepts

- "*data*" is defined as a body of facts or figures, which have been gathered systematically for one or more specific purposes
    - data can exist in the forms of
        - linguistic expressions (e.g. name, age, address, date, ownership)
        - symbolic expressions (e.g. traffic signs)
        - mathematical expressions (e.g. $E = mc^2$)
        - signals (e.g. electromagnetic waves)

- "*information*" is defined as data which have been processed into a form that is meaningful to a recipient and is of perceived value in current or prospective decision making
    - although data are ingredients of information, not all data make useful information
        - data not properly collected and organized are a burden rather than an asset to an information user
        - data that make useful information for one person may not be useful to another person
    - information is only useful to its recipients when it is
        - relevant (to its intended purposes and with appropriate level of required

detail)
        - reliable, accurate and verifiable (by independent means)
        - up-to-date and timely (depending on purposes)
        - complete (in terms of attribute, spatial and temporal coverage)
        - intelligible (i.e. comprehensible by its recipients)
        - consistent (with other sources of information)
        - convenient/easy to handle and adequately protected

- the function of an *information system* is to change "data" into "information", using the following processes (Figure 1):
    - conversion --- transforming data from one format to another, from one unit of measurement to another, and/or from one feature classification to another
    - organization --- organizing or re-organizing data according to database management rules and procedures so that they can be accessed cost-effectively
    - structuring --- formatting or re-formatting data so that they can be acceptable to a particular software application or information system
    - modeling --- including statistical analysis and visualization of data that will improve user's knowledge base and intelligence in decision making

- the concepts of "organization" and "structure" are crucial to the functioning of information systems --- without organization and structure it is simply impossible to turn data into information

1.2. Geographic data and geographic information

- geographic data are a special type of data; by "geographic", it means that
    - the data are pertinent to features and resources of the Earth, as well as the human activities based on or associated with these features and resources
    - the data are collected and used for problem solving and decision making associated with geography, i.e. location, distribution and spatial relationships within a particular geographical framework

- geographic data are different from other types of data in that
    - they are geographically referenced, i.e. they can be identified and located by coordinates
    - they are made up of a *descriptive element* (which tells what they are) and a *graphical element* (which tells what they look like, where they are found and how they are spatially related to one another)
        - the descriptive element is also commonly referred to as *non-spatial data*
        - the graphical element is also commonly referred to as *spatial data*

- geographic information is obtained by processing geographic data, the aim of which is to
    - improve the user's knowledge about the geography of the Earth's features and resources, as well as human activities associated with these features and resources
    - enable the user's to develop spatial intelligence for problem solving and decision making concerning the occurrence, utilization and conservation of the Earth's

features and resources, as well as the impacts and consequences of human activities associated with them

- because of the special nature and characteristics of geographic data, generic concepts of information organization and data structure cannot be applied directly to them
  - this unit attempts to explain the principles and methods of information organization and data structure with special reference to geographic data, particularly with respect to:
    - the organization and structure of descriptive geographic data
    - the organization and structure of graphical geographic data
    - the relationship and linkage between the descriptive and graphical elements of geographic data

1.3. The Information Domain

- an information system is designed to process data, i.e. to accept input (data), manipulate it in some way, and produce output (information) (Figure 1)

- it is also designed to process *events* --- an event represents a problem or system control which triggers data processing procedures in an information system

- the *information domain* of an information system therefore includes both data (i.e. characters, numbers, images and sound) and events (i.e. problem and control)

- there are three different components of the information domain
  - *information organization* (also referred to as *information structure*) --- the internal organization of various data and event items (see Section 2 below)
    - the design and implementation of information organization is referred to as *data structure* (see Section 3 below)
  - *information contents and relationships* --- the attributes relating to the data and the events, and the relationships with one another
    - the process of identifying information contents and relationships is known as *data modeling* in information system design (see Section 4 below)
  - *information flow* --- the ways by which data and events change as they are processed by the information system
    - the process of identifying information flow is known as *process modeling* in information system design (see Section 5 below)

- the above views of information domain provides the conceptual framework that links database management and application development in information systems
  - it signifies that information organization and data structure are not only important for the management of data, but also for the development of software applications that utilize these data

1.4. The data-oriented approach to information systems

- an information system is perceived as being made up of four components: data, technology, process (or application) and people

- the traditional approach to information system development was either *technology-oriented* or *process-oriented*
    - technology-oriented approach --- based on availability and/or functionality of hardware and software
    - process-oriented approach --- based on the desire to automate a particular business process

- the current approach to information system development is *data-oriented* or *data-driven*
    - information systems are designed and developed to process, manage and analyze data in support of the business objectives of an organization
    - of the four components of information systems, data are most stable
        - computer technology is evolving very rapidly
            - consider the advances in computer hardware and software in the last several years
            - there is always a risk of using newest technologies which have not been fully market-tested
        - processes may change due to changing business objectives, customer requirements, modes of service delivery and available tools and technology
            - consider the changes in bank transactions (depositing and withdrawing money) that have occurred in the last several years
            - process-oriented applications have relatively short life span and frequent re-development is very costly
        - particular business functions always require the same data for operation and decision-making purposes despite changes in technology and process
            - for example, bank transactions use the same data no matter how they are done (over the counter or using an automated telling machine)
    - of the four components of information systems, data are most expensive to acquire
        - in many projects, the collection of data accounts for half or more of the capital investment
        - it is natural to make use of the most stable and expensive component to drive information system design and development in order to maximize the return from capital investment

- a data-oriented approach to information system is characterized by
    - managing data as a valuable corporate resource in the same way financial, technical and human resources are managed
        - this is the basis of the concept of *information resource management* (IRM)
    - sharing data among different users or user groups
        - this helps maximize the cost-benefit ratio of the capital investments on information systems
    - data-centric strategy in the acquisition of hardware and software
        - specifications of hardware and software must be able to meet data requirements, but not to change data requirements in order to suit the characteristics or functionality of hardware and software
    - data-driven application development
        - software applications are designed to enable the effective and efficient use

of data in business operation and decision making
        - the use of information systems is always used by organizations as an opportunity to re-engineer business, i.e. to change the philosophy and ways of running a business

- data orientation does not mean that every user is involved in information organization and data structure
    - ensuring that information organization and data structure meet the business needs of an organization is the responsibility of a small team of technical staff under the leadership of a *database administrator*
        - the database administration team defines an organization's information organization by carrying out detailed user requirement studies
        - representatives of end users assist in defining information organization by taking part in user requirement studies
    - system developers design and build software applications without the need to worry about information organization and data structure, i.e. they develop applications on the basis of existing or accepted data structures
        - however, system developers may also assist in defining information organization by taking part in user requirement studies
    - information organization and data structure are transparent to end users, i.e. they can use software applications without the need to know anything about data structure

- however, this does not imply that information organization and data structure are trivial considerations in information system
    - information organization and data structure reflect the users' requirements
        - many projects fail because of the lack of understanding of information organization and data structure, but not because of the lack of capability of technology
        - identification of users' requirements, which forms the foundation of good design and correct specification of information organization and data structure, is always the most important step in information system development

- the ultimate goal of information organization and data structure is to create the necessary technical environment that allows the development of information systems which are
    - cost-effective to implement --- by the ability to use shared data and possibly applications by users in different organizations
    - flexible to build --- by permitting the addition or removal of applications, in response to changing needs and objectives of the information users, without affecting existing data structure
    - easy to use --- by eliminating the need of the regular users to worry about the structure of the data

---

2. Information Organization

- information organization can be understood from four perspectives:
  - a data perspective
  - a relationship perspective
  - an operating system (OS) perspective
  - an application architecture perspective

2.1. The data perspective of information organization

- the information organization of geographic data must be considered in terms of their descriptive elements and graphical elements because
  - these two types of data elements have distinctly different characteristics
  - the have different storage requirements
  - they have different processing requirements

2.1.1. Information organization of descriptive data

- for descriptive data, the most basic element of information organization is called a *data item* (Figure 2a)
  - a data item represents an *occurrence* or *instance* of a particular characteristic pertaining to an entity (which can be a person, thing, event or phenomenon)
    - it is the smallest unit of stored data in a database, commonly referred to as an *attribute*
      - in database terminology, an attribute is also referred to as a *stored field*
    - the value of an attribute can be in the form of a number (integer or floating-point), a character string, a date or a logical expression (e.g. *T* for 'true' or 'present''; *F* for 'false' or 'absent')
    - some attributes have a definite set of values known as *permissible values* or *domain of values* (e.g. age of people from 1 to 150; the categories in a land use classification scheme; and the academic departments in a university)

- a group of related data items form a *record* (Figure 2b)
  - by related data items, it means that the items are occurrences of different characteristics pertaining to the same person, thing, event or phenomenon (e.g. in a forest resource inventory, a record may contain related data items such as stand identification number, dominant tree species, average height and average breast height diameter)
  - a record may contain a combination of data items having different types of values (e.g. in the above example, a record has two character strings representing the stand identification number and dominant tree species; an integer representing the average tree height rounded to the nearest meter; and a floating-point number representing the average breast height diameter in meters)
    - in database terminology, a record is always formally referred to as a *stored record*
    - in relational database management systems, records are called *tuples*

- a set of related records constitutes a *data file* (Figure 2c)

- by related records, it means that the records represent different occurrences of the same type or class of people, things, events and phenomena
  - a data file made up of a single record type with single-valued data items is called a *flat file* (Figure 3a)
  - a data file made up of a single record type with nested repeating groups of items forming a multi-level organization is called a *hierarchical file* (Figure 3b)
- a data file is individually identified by a *filename*
- a data file may contain records having different types of data values or having a single type of data value
  - a data file containing records made up of character strings is called a *text file* or *ASCII file*
  - a data file containing records made up of numerical values in binary format is called a *binary file*
- in data processing literature, collections of data items or records are sometimes referred to by other terms other than "data file" according to their characteristics and functions
  - an *array* is a collection of data items of the same size and type (although they may have different values)
    - a one-dimensional array is called a *vector*
    - a two-dimensional array is called a *matrix*
  - a *table* is a data file with data items arranged in rows and columns
    - data files in relational databases are organized as tables
    - such tables are also called *relations* in relational database terminology
  - a *list* is a finite, ordered sequence of data items (known as *elements*)
    - by "ordered", it means that each element has a position in the list
    - an ordered list has elements positioned in ascending order of values; while an unordered list has no permanent relation between element values and position
    - each element has a data type
    - in the simple list implementation, all elements must have the same data type but there is no conceptual objection to lists whose elements have different data types
  - a *tree* is a data file in which each data item is attached to one or more data items directly beneath it (Figure 4)
    - the connections between data items are called *branches*
    - trees are often called *inverted trees* because they are normally drawn with the root at the top
    - the data items at the very bottom of an inverted tree are called *leaves*; other data items are called *nodes*
    - a *binary tree* is a special type of inverted tree in which each element has only two branches below it
  - a *heap* is a special type of binary tree in which the value of each node is greater than the values of its leaves
    - heap files are created for sorting data in computer processing --- the *heap sort algorithm* works by first organizing a list of data into a heap

        - a *stack* is a collection of *cards* in Apple Computer's *Hypercard* software system

- the concept of *database* is the approach to information organization in computer-based data processing today
    - a database is defined as an automated, formally defined and centrally controlled collection of persistent data used and shared by different users in an enterprise (Date, 1995 and Everest, 1986)
        - above definition excludes the informal, private and manual collection of data
        - "centrally controlled" does not mean "physically centralized" --- databases today tend to be physically distributed in different computer systems, at the same or different locations
        - a database is set up to serve the information needs of an organization
        - data sharing is key to the concept of database
        - data in a database are described as "permanent" in the sense that they are different from "transient" data such as input to and output from an information system
            - the data usually remain in the database for a considerable length of time, although the actual content of the data can change very frequently
    - the use of database does not mean the demise of data files
        - data in a database are still organized and stored as data files
        - the use of database represents a change in the perception of data, the mode of data processing and the purposes of using the data (Table 1), rather than physical storage of the data
    - databases can be organized in different ways known as *database models*
        - the three conventional database models are: *relational*, *network* and *hierarchical*
            - relational --- data are organized by records in relations which resemble a table (Figure 5a) (See Section 3.2.1 for further explanation)
            - network --- data are organized by records which are classified into record types, with 1:n pointers linking associated records (Figure 5b)
            - hierarchical --- data are organized by records on a parent-child one-to-many relations (Figure 5c)
        - the emerging database model is *object-oriented*
            - data are uniquely identified as individual objects that are classified into object types or classes according to the characteristics (attributes and operations) of the object (Figure 5d) (See Section 3.2.2 for further explanation)

2.1.2. Information organization of graphical data

- for graphical data, the most basic element of information organization is called a *basic graphical element*
    - there are three basic graphical elements (Figure 6):

point
    - *line*, also referred to as *arc*
    - *polygon*, also referred to as *area*
  - these basic graphical elements can be individually used to represent geographic features or entities
    - for example: point for a well; line for a road segment and polygon for a lake)
  - they can also be used to construct complex features
    - for example: the geographic entity "Hawaii" on a map is represented by a group of polygons of different sizes and shapes

- the method of representing geographic features by the basic graphical elements of points, lines and polygon is said to be the *vector method* or *vector data model*, and the data are called *vector data*
  - related vector data are always organized by *themes*, which are also referred to as *layers* or *coverages*
    - examples of themes: geodetic control, base map, soil, vegetation cover, land use, transportation, drainage and hydrology, political boundaries, land parcel and others
  - for themes covering a very large geographic area, the data are always divided into *tiles* so that they can be managed more easily
    - a tile is the digital equivalent of an individual map in a map series
    - a tile is uniquely identified by a file name
  - a collection of themes of vector data covering the same geographic area and serving the common needs of a multitude of users constitutes the *spatial component* of *a geographical database*
  - the vector method of representing geographic features is based on the concept that these features can be can be identified as discrete entities or objects
    - this method is therefore based on the *object view of the real world* (Goodchild, 1992)
    - the object view is the method of information organization in conventional mapping and cartography

- graphical data captured by imaging devices in remote sensing and digital cartography (such as multi-spectral scanners, digital cameras and image scanners) are made up of a matrix of picture elements (pixels) of very fine resolution
  - geographic features in such form of data can be visually recognized but not individually identified in the same way that geographic features are identified in the vector method
  - they are recognizable by differentiating their spectral or radiometric characteristics from pixels of adjacent features
    - for example, a lake can be visually recognized on a satellite image because the pixels forming it are darker than those of the surrounding features; but the pixels forming the lake are not identified as a single discrete geographic entity, i.e. they remain individual pixels
    - similarly, a highway can be visually recognized on the same satellite image because of its particular shape; but the pixels forming the highway do not constitute a single discrete geographic entity as in the case of vector data

- the method of representing geographic features by pixels is called the *raster method* or *raster data model*, and the data are described as *raster data*
  - the raster method is also called the *tessellation method*
  - a raster pixel is usually a square grid cell but there are there are several variants such as triangles and hexagons (Peuquet, 1991)
  - a raster pixel represents the generalized characteristics of an area of specific size on or near the surface of the Earth
    - the actual ground size depicted by a pixel is dependent on the resolution of the data, which may range from smaller than a square meter to several square kilometers
  - raster data are organized by themes, which is also referred to as layers
    - for example, a raster geographic database may contain the following themes: bed rock geology, vegetation cover, land use, topography, hydrology, rainfall, temperature
  - raster data covering a large geographic area are organized by *scenes* (for remote sensing images) of by *raster data files* (for images obtained by map scanning)
  - the raster method is based on the concept that geographic features are represented as surfaces, regions or segments
  - this method is therefore based on the *field view of the real world* (Goodchild, 1992)
  - the field view is the method of information organization in image analysis systems in remote sensing and geographic information systems for resource- and environmental-oriented applications

- in the past, the vector and raster methods represented two distinct approaches to information systems
  - they were based on different concepts of information organization and data structure
  - they used different technologies for data input and output

- recent advances in computer technologies allow these two types of data to be used in the same applications
  - computers are now capable of converting data from the vector format to the raster format (rasterization) and vice versa (vectorization)
  - computers are now able to display vector and raster simultaneously
  - the old debate on the usefulness of these two approaches to information organization does not seem to be relevant any more
  - vector and raster data are largely seen as complimentary to, rather than competing against, one another in geographic data processing

2.2. The relationship perspective of information organization

- relationships represent a important concept in information organization --- it describes the logical association between entities
  - relationships can be *categorical* or *spatial*, depending on whether they describe location or other characteristics

2.2.1. Categorical relationships

- categorical relationships describe the association among individual features in a classification system
  - the classification of data is based on the concept of *scale of measurement*
  - there are four scales of measurement:
    - *nominal* --- a qualitative, non-numerical and non-ranking scale that classifies features on intrinsic characteristics
      - for example, in a land use classification scheme, polygons can be classified as industrial, commercial, residential, agricultural, public and institutional
    - *ordinal* --- a nominal scale with ranking which differentiates features according to a particular order
      - for example, in a land use classification scheme, residential land can be denoted as low density, medium density and high density
    - *interval* --- an ordinal scale with ranking based on numerical values that are recorded with reference to an arbitrary datum
      - for example, temperature readings in degrees centigrade are measured with reference to an arbitrary zero (i.e. zero degree temperature does not mean no temperature)
    - *ratio* --- an interval scale with ranking based on numerical values that are measured with reference to an absolute datum
      - for example, rainfall data are recorded in mm with reference to an absolute zero (i.e. zero mm rainfall mean no rainfall)

- categorical relationships based on ranking are hierarchical or taxonomic in nature
  - this means that data are classified into progressively different levels of detail
    - data in the top level are represented by a limited broad basic categories
    - data in each basic category are then classified into different sub-categories, which can be further classified into another level if necessary
  - the classification of descriptive data is typically based on categorical relationships (Figure 7)

2.2.2. Spatial relationships

- spatial relationships describe the association among different features in space
  - spatial relationships are visually obvious when data are presented in the graphical form
  - however, it is difficult to build spatial relationships into the information organization and data structure of a database
    - there are numerous types of spatial relationships possible among features (Table 2)
    - recording spatial relationships implicitly demands considerable storage space
    - computing spatial relationships on-the-fly slows down data processing particularly if relationship information is required frequently

there are two types of spatial relationships (Figure 8)
  - *topological* --- describes the property of adjacency, connectivity and containment of contiguous features
  - *proximal* --- describes the property of closeness of non-contiguous features

- spatial relationships are very important in geographical data processing and modeling
  - the objective of information organization and data structure is to find a way that will handle spatial relationships with the minimum storage and computation requirements

2.3. The operating system (OS) perspective of information organization

- from the operating system perspective, information is organized in the form of *directories*
  - directories are a special type of computer files used to organize other files into a hierarchical structure (Figure 9)
    - directories are also referred to as *folders*, particularly in systems using graphical user interfaces
  - a directory may also contain one of more directories
    - the topmost directory in a computer is called the root directory
    - a directory that is below another directory is referred to as a *sub-directory*
    - a directory that is above another directory is referred to as a *parent directory*
  - directories are designed for bookkeeping purposes in computer systems
    - a directory is identified by a unique directory name
    - computer files of the same nature are usually put under the same directory
    - a data file can be accessed in a computer system by specifying a *path* that is made up of the device name, one or more directory names and its own file name
      - for example: c:\project101\mapdata\basemap\nw2367.dat
  - the concept of *workspace* used by many geographic information system software packages is based on the directory structure of the host computer
    - a workspace is a directory under which all data files relating to a particular project are stored (Figure 10)

2.4. The application architecture perspective of information organization

- computer applications nowadays tend to be constructed on the *client/server* systems architecture

- client/server is primarily a relationship between processes running in the same computer or, more commonly, in separate computers across a telecommunication network (Figure 11)
  - the *client* is a process that requests services
    - the dialog between the client and the server is always initiated by the client
    - a client can request services from many servers at the same time
  - the *server* is a process that provides the service

        a server is primarily a passive service provider
        - a server can service many clients at the same time

- there are many ways of implementing a client/server architecture but from the perspective of information organization, the following five are most important
    - *file servers* --- the client requests specific records from a file; and the server returns these records to the client by transmitting them across the network
    - *database servers* --- the client sends *structured query language* (SQL) requests to the server; the server finds the required information by processing these requests and then passes the results back to the client
    - *transaction servers* --- the client invokes a remote procedure that executes a transaction at the server side; the server returns the result back to the client via the network
    - *Web server* --- communicating interactively by the *Hypertext Transfer Protocol* (HTTP) over the Internet, the Web server returns documents when clients ask for them by name
    - *groupware servers* --- this particular type of servers provides a set of applications that allow clients (and their users) to communicate with one another using text, images, bulletin boards, video and other forms of media

- from the application architecture perspective, the objective of information organization and data structure is to develop a data design strategy that will optimize system operation by
    - balancing the distribution of data resources between the client and the server
        - databases are typically located on the server to enable data sharing by multiple users
        - static data that are used for reference are usually allocated to the client
    - ensuring the logical allocation of data resources among different servers
        - data that are commonly used together should be placed in the same server
        - data that have common security requirements should be placed in the same server
        - data intended for a particular purpose (file service, database query, transaction processing, Web browsing or groupware applications) should be placed in the appropriate server
    - standardizing and maintaining metadata (i.e. data about data) to facilitate the search for the availability and characteristics of existing data

---

3. Data Structure

3.1. Levels of data abstraction

- as noted in Section 1.3, information organization is concerned with the internal organization of data
    - it represents the user's view of data, i.e. conceptualization of the real world
    - it is the lowest level of data abstraction, which can be done with or without any intent for computer implementation

    - it is expressed in terms of *data models* (Peuquet, 1991) (Figure 12)
        - note the differences between "data models" and "database models"
            - the vector and raster methods of representing the real world as explained in Section 2.1.2 above are "data models"
            - the relational, network, hierarchical and object-oriented databases are "database models" --- they are the software implementation of data models

- *data structure* represents a higher level of data abstraction than information organization in the sense that it is concerned with the design and implementation of information organization
    - it represents the human implementation-oriented view of data
    - it is expressed in terms of database models
        - this implies that data structure is software-dependent but hardware is not yet a consideration

- data structure forms the basis for the next level of data abstraction in information system: *file structure* or *file format*
    - file structure is the hardware implementation-oriented view of data
    - it reflects the physical storage of the data on some specific computer media such as magnetic tapes or hard disk
        - this implies that file structure is hardware-dependent

3.2. Descriptive data structures

- descriptive data structures describe the design and implementation of the information organization of non-spatial data

- as most commercial implementations of information systems today are based on the relational and object-oriented database models, we explain the data structures of these models in the following two sections

3.2.1. Relational data structure

- the relational data structure is the table which is formally called a relation (Figure 13)
    - a relation is a collection of tuples that correspond to the rows of the table
        - the number of tuples in a relation is called the *cardinality*
    - a tuple is made up of attributes that correspond to the columns of the table
        - the number of attributes in a tuple is called the *degree*
    - each relation has a unique identifier called the *primary key*
        - the primary key is a column or combination of columns that at any given time has no identical values in any two rows
            - this means that the values of each row of the primary key are always unique
            - this allows the use of the primary keys to relate data in different tables in data processing (Figure 13)
                - the primary keys in those tables are called *foreign keys*

in order to enforce database integrity, relations are always *normalized*

- normalization is built on the concept of *normal form*
- a relation is said to be in a certain normal form if it satisfies a prescribed set of conditions (Date, 1995)
- as a minimum, a relation in the relational database has to satisfy the conditions of the first, second and third normal forms
  - *first normal form* (1NF) --- a relation is said to be in 1NF if and only if its tuples contain no repeating attributes (i.e. there must not be multiple values for a single entity which might theoretically result from multiple sampling at a particular location)
  - *second normal form* (2NF) --- a relation is said to be in 2NF if it satisfies the condition for 1NF and if every non-key attribute is irreducibly dependent on the primary key
  - *third normal form* (3NF) --- a relation is said to be in 3NF if it satisfies the condition for 2NF and the non-key attributes are mutually independent

3.2.2. Object-oriented data structure

- unlike the relational data structure, there is not a formalized object-oriented data structure
  - this means that different object-orientation implementations have different data structures

- however, object-oriented data structure can be explained in generic terms using the concepts of *object identify*, *object structure* and *type constructors* (Elmasri and Navathe, 1994)
  - the concept of object identity
    - each object in an object-oriented database is provided a unique system-generated *object identifier* (OID)
    - the OID is for internal reference by the system and is therefore transparent to the user
    - the OID is *immutable*, i.e. its value remains unchanged
      - even when a particular object is removed from the database, its OID will never be assigned to any new object
  - the concept of object structure
    - the concept of object structure allows complex objects to be constructed from simple objects
    - each object is viewed as a triple (i, c, v) where

      i = the object's unique identifier (OID)
      c = a constructor (which indicates how the object value is constructed)
      v = object value

      - different object-oriented systems use different constructors, including: atom, tuple, set, list and array
      - an object value v is interpreted on the basis of the value of the

constructor c in the triple (i, c, v) that represents the object
- if c = atom, then v is an atomic value (i.e. it is an indivisible value)
- if c = tuple, then v is a tuple containing one or more attributes with their respective OIDs
- if c = set, then v is a set of object identifiers (OIDs) for a set of objects of the same type
- if c = list, then v is an ordered list of OIDs of the same type
- if c = array, then v is an array of OIDs of the same type

- the concept of type constructors
  - a type constructor is used by an *object-oriented definition language* (OODDL) to define the data structure for an *object-oriented database schema* (Figure 14)

3.3. Graphical data structures

3.3.1. Raster data structure

- in the raster data structure space is subdivided into regular grids of square grid cells or other forms of polygonal meshes known as picture elements (pixels) (Figure 15)
  - the location of each cell is defined by its row and column numbers
  - the area that each cell represents defines the spatial resolution of the data
  - the position of a geographic feature is only recorded to the nearest pixel
  - the value stored for each cell indicates the types of the object, phenomenon or condition that is found in that particular location
    - different types of values can be coded: integers, real numbers and alphabets
    - integer values often act as code numbers, which are referenced to names in an associated table (called the *look-up table*) or legend
  - different attributes at the same cell location are stored as separate themes or layers
    - for example, raster data pertaining to the soil type, forest cover and slope covering the same area are stored separately in a soil type theme, a forest cover theme and a slope theme

- there are several variants to the regular grid raster data structure, including: *irregular tessellation* (e.g. triangulated irregular network (TIN)), *hierarchical tessellation* (e.g. quad tree) and *scan-line* (Peuquet, 1991)

3.3.2. Vector data structure

- there are many implementations of vector data structures, including:
  - *spaghetti* --- a direct line-for-line unstructured translation of the paper map (Figure 16)
    - this structure has very limited practical use
    - it is usually an interim data structure for map digitizing
  - *hierarchical* --- a vector data structure developed to facilitate data retrieval by separately storing points, lines and areas in a logically hierarchical manner

(Figure 17)
    - *topological* --- a vector data structure that aims at retaining spatial relationship by explicitly storing adjacency information (Figure 18)
        - the basic logical feature for line and area coverage is a straight line segment
        - each individual line segment is defined by the coordinates of its end points called *nodes*
        - topological information is stored by recording
            - the from-node and to-node of each line segment
            - the left-polygon and right-polygon (in the direction of the from-node to the to-node) of each line segment

3.4. The georelational data structure

- the georelational data structure was developed to handle geographic data
    - it allows the association between spatial (graphical) and non-spatial (descriptive) data
    - it is the data structure used by many vector-based GIS software packages
    - both spatial and non-spatial data are stored in relational tables
        - point, line and polygon data are stored in separate *feature attribute tables* (FAT) (Figure 19)
            - in the FAT, each entity is assigned a unique feature identifier (FID)
            - topological information is explicitly stored by employing a method similar to the topological data structure described above
        - non-spatial data are stored in relational tables
    - entities in the spatial and non-spatial relational tables are linked by the common FIDs of entities (Figure 20)

---

4. Data Modeling

- data modeling is the process of defining real world phenomena or geographic features of interest in terms of their characteristics and their relationships with one another
    - it is concerned with different phases of work carried out to implement information organization and data structure

- there are three steps in the data modeling process, resulting in a series of progressively formalized data models as the form of the database becomes more and more rigorously defined
    - *conceptual data modeling* --- defining in broad and generic terms the scope and requirements of a database
    - *logical data modeling* --- specifying the user's view of the database with a clear definition of attributes and relationships
    - *physical data modeling* --- specifying internal storage structure and file organization of the database

- data modeling is obviously closely related to the three levels of data abstraction in database design as noted in Section 3.1 above:

- conceptual data modeling ----> data model
- logical data modeling ---------> data structure
- physical data modeling -------> file structure

4.1. Conceptual data modeling

- *entity-relationship* (E-R) modeling is probably the most popular method of conceptual data modeling
  - it is sometimes referred to as a method of *semantic data modeling* because it used a human language-like vocabulary to describe information organization
  - it involves four aspects of work:
    - identifying entities
      - an entity is defined as a person, a place, an event, a thing, etc.
    - identifying attributes
    - determining relationships
    - drawing an *entity-relationship diagram* (E-R diagram) (Figure 21)

4.2. Logical data modeling

- logical data modeling is a comprehensive process by which the conceptual data model is consolidated and refined
  - the proposed database is reviewed in its entirety in order to identify potential problems such as
    - irrelevant data that will not be used
    - omitted or missing data
    - inappropriate representation of entities
    - lack of integration between various parts of the database
    - unsupported applications
    - potential additional cost to revise the database
  - the end product of logical data modeling is a *logical schema*
    - the logical schema is developed by mapping the conceptual data model (such as the E-R diagram) to a software-dependent design document (Figure 22)

4.3. Physical data modeling

- physical data modeling is the database design process by which the actual tables that will be used to store the data are defined in terms of
  - data format --- the format of the data that is specific to a database management system (DBMS)
  - storage requirements --- the volume of the database
  - physical location of data --- optimizing system performance by minimizing the need to transmit data between different storage devices or data servers
- the end product of physical data modeling is a *physical schema* (Figure 23)
  - a physical schema is also variably known as *data dictionary*, *item definition table*, *data specific table* or *physical database definition*
  - it is both software- and hardware specific

- this means the physical schemas for different systems look different from one another

---

5. Process Modeling

- process modeling is the process-oriented approach, as opposed to the data-oriented approach, of information system design
  - the objective is to identify the processes that the information system will perform
  - it also aims at identifying how information is transformed from one process to another
  - the end product of process modeling is a *data flow diagram* (DFD)
    - this implies that process modeling is by no means only concerned with process, it also deals with information organization and data structure

- in the context of information system design, process modeling is one of the methods of *structured business function decomposition* used to determine user requirements in conceptual modeling
  - DFD is the principal modeling tool
  - a DFD is constructed using four basic symbols to represent *process*, *data stores*, *entities* and *data flow* in a business function (Figure 24)
    - process --- it represents the transformation of data as they flow through the system: data flow into a process, are changed, and then flow out to another process or a data store
    - entity --- the basic definition of an entity is similar to that for E-R modeling and it represents the initial source and final destination of data in a DFD
    - data store --- a temporary or permanent holding area for data
    - data flow --- the connection between processes and data stores along which individual entities or collection of entities flow

- process modeling is a top-down analysis and design method
  - it results in a hierarchy of DFDs that represent a general-to-detail decomposition of processes (Figure 25)
  - a top level DFD, called the Level-0 DFD, typically contains a single process or a small number of processes that describes a business from a global perspective
  - this DFD is then decomposed into lower levels of DFDs (i.e. Level-1, Level-2, etc.) that provide progressively more detailed breakdown of business processes
  - the final DFD is used as the basis for information organization and data structure in the process-oriented approach to information system development

---

6. Summary

- this unit represents an overview, rather than a detailed explanation, of the principles and methods of information organization and data structure
  - the aim is to provide students with an articulate view of information organization

from the conceptualization, through design and specification, to the practical implementation of data and file structures in information science and management
    - this enables students to understand how different processes in information system development, such as data modeling, database design and application development, are related to one another
- information organization refers to the internal organization of data and event items in information systems
- information organization is a key consideration in today's data-oriented approach to system design and development; it is crucial to the functioning of information systems
- information organization is largely conceptual in nature, and can be understood from four interrelated perspectives: data, relationship, operating system and application architecture
- data structure is the design and implementation of information organization; it is the intermediate step of work between conceptual database design and the practical implementation of file structures
- the identification of entities, attributes and relationships for data structure can be carried out either by data modeling or process modeling

---

7. Review and Study Questions

1. The following three lines of figures have been extracted from a computer file:

   00713344 5000 7.50 1998 12 31 000999999999999
   23112410 0500 7.50 1999 11 01 000999999999999
   33132211 8000 8.00 2001 06 30 000999999999999

   Are these data or information? Explain why.

2. Explain the importance of information organization and data structure to the functioning of information systems.

3. Information organization and data structure are key considerations in information system development. Who are the people responsible for identifying, specifying and implementing an organization's requirements for information organization and data structure?

4. Explain the differences between geographic data and other types of data from the perspective of information organization and data structure.

5. List the characteristics of the database approach as opposed to the conventional data file approach to data processing.

6. Explain the difference between a "data model" and a "database model"

7. Define "categorical relationship" and "spatial relationship". Explain why spatial relationships are more difficult than categorical relationships to implement in data structure.

8. Information systems are now mostly based on the client/server architecture. Explain the impact of this particular architecture on information organization in system implementation.

9. What is a relation in the context of data structure? List the characteristics of a relation in terms of data structure.

10. What is an object in the context of data structure? How is the data structure for an object-oriented database schema constructed?

11. Explain the relationships among conceptual, logical and physical modeling in database design.

12. What is a data flow diagram? Explain how a data flow diagram can be used in connection with information organization and data structure.

---

8. References

Date, C.J. (1995) *An Introduction to Database Systems* (6th ed.) Addison-Wesley, Reading, MA.

Elmasri, R. and Navathe, S.B. (1994) *Fundamentals of Database Sys*tems. Addison-Wesley, Menlo Park, CA.

Everst, G.C. (1986*) Data Management: Objectives, System Functions and Administration*, McGraw-Hill, New York.

Goodchild, M.F. (1992) Geographic Data Modeling. *Computers and Geosciences*. Vol. 18, No. 4, pp. 401-408.

Peuquet, D.J. (1991) Methods for Structuring Digital Cartographic Data in a Personal Computer Environment. In *Geographic Information Systems: The Microcomputer and Modern Cartography* by Taylor, D.R.F. (ed.), Pergamon Press, Oxford.

Pressman, R.S. (1997) *Software Engineering: A Practitioner's Approach* (4th ed.) McGraw-Hill, New York.

Citation

To reference this material use the appropriate variation of the following format:

> Albert K. Yeung. (1998) Data Organization and Structure, *NCGIA Core Curriculum in GIScience*, http://www.ncgia.ucsb.edu/giscc/units/u051/u051.html, posted October 15, 1998.

Last revised: October 12, 1998.

Unit 051 - Information Organization and Data Structure

Table of Contents

Advanced Organizer
- Topics covered in this unit
- Intended learning outcomes
- Instructors' Notes
- Metadata and Revision History

1. Definitions and Terminology

- 1.1 Data and information
- 1.2 Geographic data and geographic information
- 1.3 The information domain
- 1.4 the data-oriented approach to information systems

2. Information Organization

- 2.1 The data perspective of information organization
  - 2.1.1 Informaiton organization and descriptive data
  - 2.1.2 Information organization and graphical data
- 2.2 The relationship perspective of information organization
  - 2.2.1 Categorical relationships
  - 2.2.2 Spatial relationships
- 2.3 The operating system perspective of information organization
- 2.4. The application architecture perspective of information organization

3. Data structure

- 3.1 Levels of data abstraction
- 3.2 Descriptive data structures
  - 3.2.1 Relational data structure
  - 3.2.2 Object-oriented data structure
- 3.3 Graphical data structures
  - 3.3.1 Raster data structure
  - 3.3.2 Vector data structure
- 3.4 The georelational data structure

4. Data modeling

- 4.1 Conceptual data modeling
- 4.2 Logical data modeling
- 4.3 Physical data modeling

5. Process modeling

6. Summary

7. Exam and study questions

8. References

Citation

Back to the Unit

Unit 051 - Information Organization andData Structure

Metadata and Revision History

1. About the main contributors

- Albert K.Yeung, Ontario Ministry of Northern Development and Mines, Canada

2. Details about the file

- unit title
  - Information Organization andData Structure
- unit key number
  - 051

3. Key words

4. Index words

5. Prerequisite units

6. Subsequent units

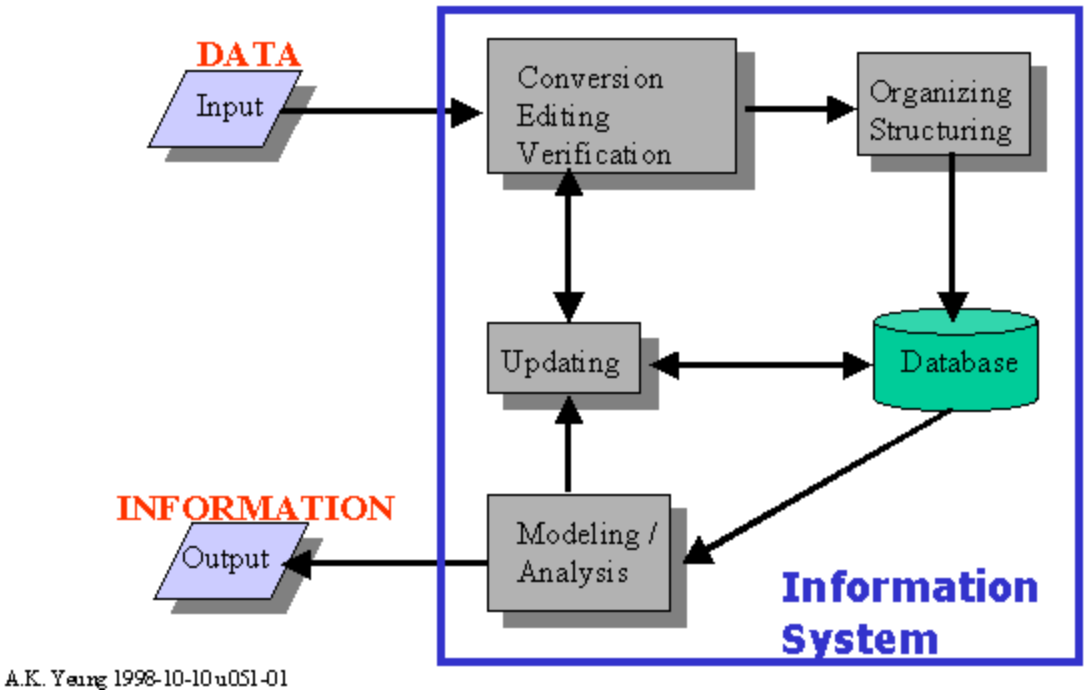
7. Other contributors to this unit

8. Revision history

- Created October 10, 1998
- Revised by author October 12, 1998
- Posted to GISCC October 28, 1998

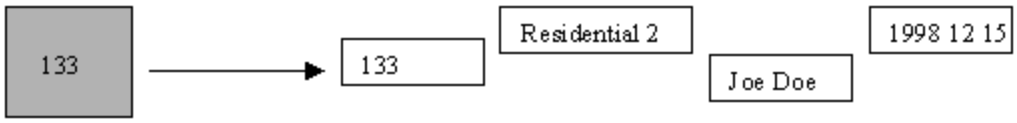
Back to the Unit.

Figure 1: Changing data into information in an information system



A.K. Yeung 1998-10-10 u051-01

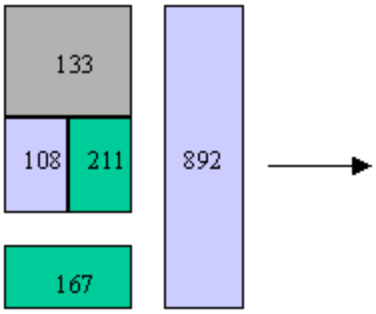
Figure 2: Data item, record, data file



(a) Data items pertaining to a land parcel

100 12113	Residential 2	Joe Doe	1998 12 15

(b) A Record of data items



133	Residential 2	Joe Doe	1998 12 15
108	Residential 3	K Jones	1987 10 10
211	Residential 1	B Blower	1996 03 15
167	Residential 2	G Smith	1945 12 02
892	Commercial 3	B King	1998 12 14

(b) A table of records

A.K. Yeung 1998-10-10 u51-02

Figure 3: Flat file and hierarchical file

A flat file

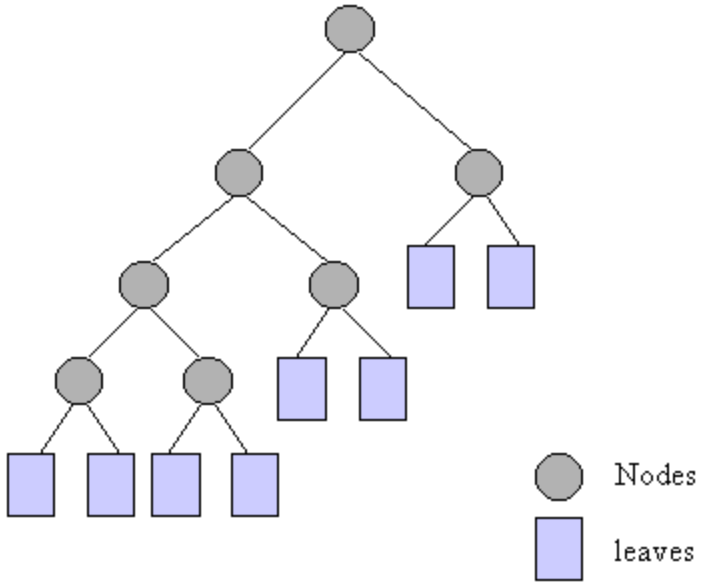
Census District	Population	No. Of Households	Medium Income ($/yr)
C13	1235	402	35,000
C14	1225	385	39,000
D18	1812	461	45,000

A hierarchical file

Census District	Population			No. Of Households		Medium Income ($/yr)	
	1985	1990		1985	1990	1985	1990
		M	F				
C13	1235	731	688	402	453	35,000	37,500
C14	1225	778	686	385	733	39,000	42,000
D18	1812	988	975	461	632	45,000	48,000

A.K. Yeung 1998-10-10 u51-03

Figure 4: The tree data structure

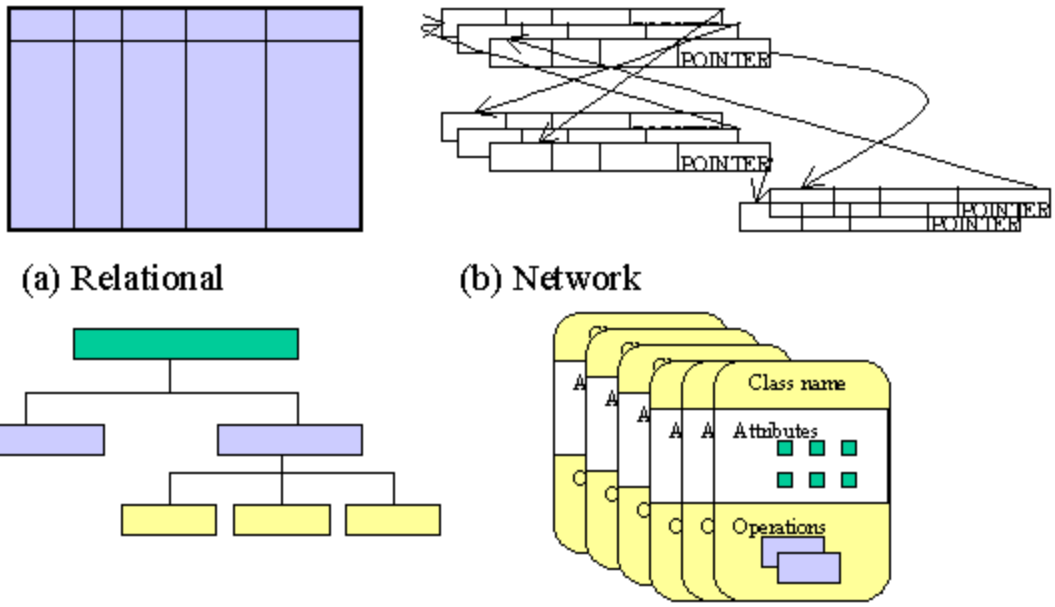


A.K. Yeung 1998-10-10 u51-04

Table 1: Distinction between a data file and a database

Characteristics of a data file	***Characteristics of a database***
A collection of records usually of the same data type and format description	A collection of interrelated records, organized in one or more data files, that may have different data types and format descriptions
Data file processing is usually associated with computer programming that aims at solving a particular problem, i.e. it stops when an answer is obtained	Database processing is always associated with database management systems that aim at solving the operation or production needs of an organization, i.e. it involves routine, largely repetitive applications executed over and over again
Mainly used in support of the information need of an *ad hoc* application	Mainly used in support of the day to day operation of business (transaction processing) but increasingly used in decision support (management decision making)

Figure 5: Database models



(a) Relational

(b) Network

(c) Hierarchical

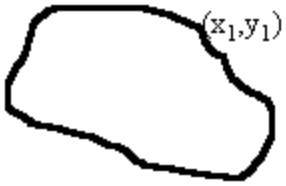
(d) Object-oriented

A.K. Yeung 1998-10-10 u51-05

Figure 6: Basic graphical elements

A point (x,y)

A line $(x_1,y_1; x_2,y_2;x_3,y_3; \ldots x_{n-1},y_{n-1};x_n,y_n)$
x_1,y_1 and x_n,y_n are called nodes; the other points are called vertices



An area $(x_1,y_1; x_2,y_2;x_3,y_3; \ldots x_n,y_n;x_1,y_1)$
x_1,y_1 is called a node; the other points are called vertices

A.K. Yeung 1998-10-10 u51-06

Figure 7: Example of a classification scheme of descriptive data

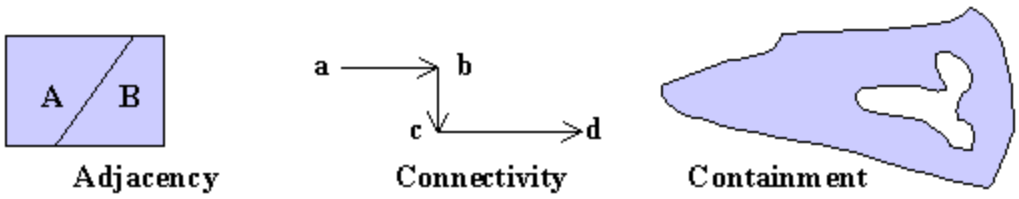
Level I	Level II
1 Urban or built-up land	11 Residential 12 Commercial and services 13 Industrial 14 Transportation ....
2 Agricultural land	21 Cropland and pasture 22 Orchards, groves, vineyards, nurseries 23 Confined feeding operations .....
3 Rangeland	31 Herbaceous rangeland 32 Shrub and brush rangeland ....
4 Forest land	41 Deciduous forest land 42 Evergreen forest land 43 Mixed forest land ....

A.K. Yeung 1998-10-10 u51-07

Table 2: Point-line-area relationship matrix

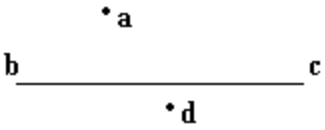
	Point	*Line*	*Area*
Point	Is nearest to Is neighbor of	Ends at Is nearest to Lies on	Is within Outside of Can be seen from
Line		Crosses Joins Flows into Comes within Is parallel to	Crosses Borders Intersects
Area			Overlaps Is nearest to Is adjacent to Is contained in

Figure 8: Topological and proximal relationships



Adjacency Connectivity Containment

Topological relationships

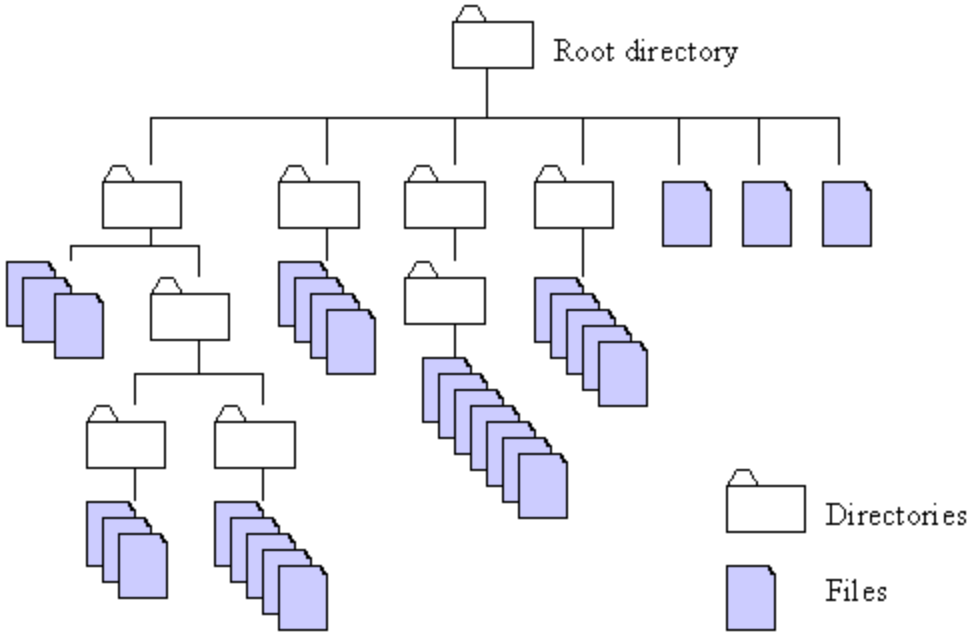


Point a is far away from line bc; point d is close to line bc

Proximal relationships

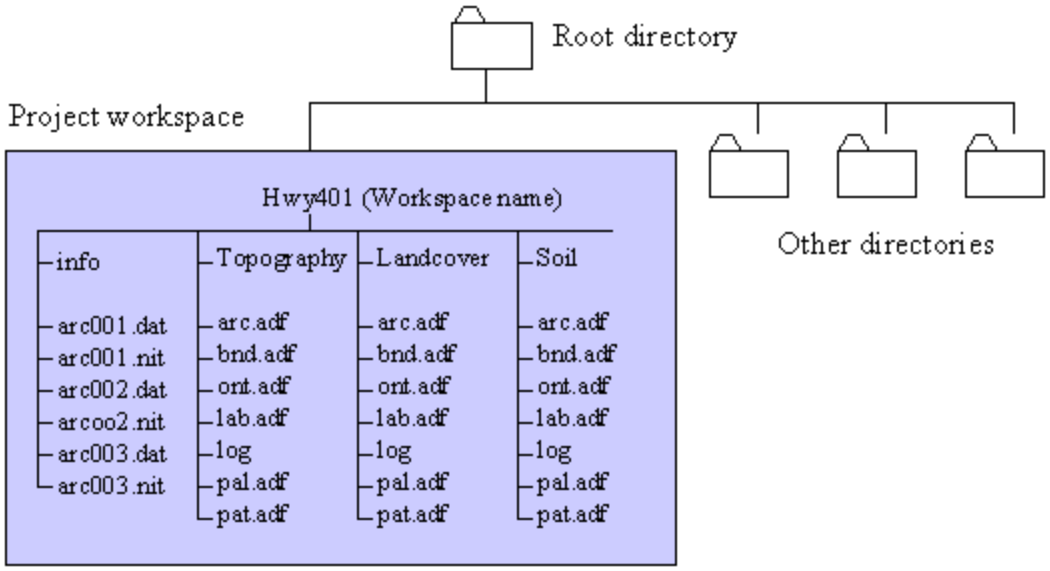
A.K. Yeung 1998-10-10 u51-08

Figure 9: Information organization by directories



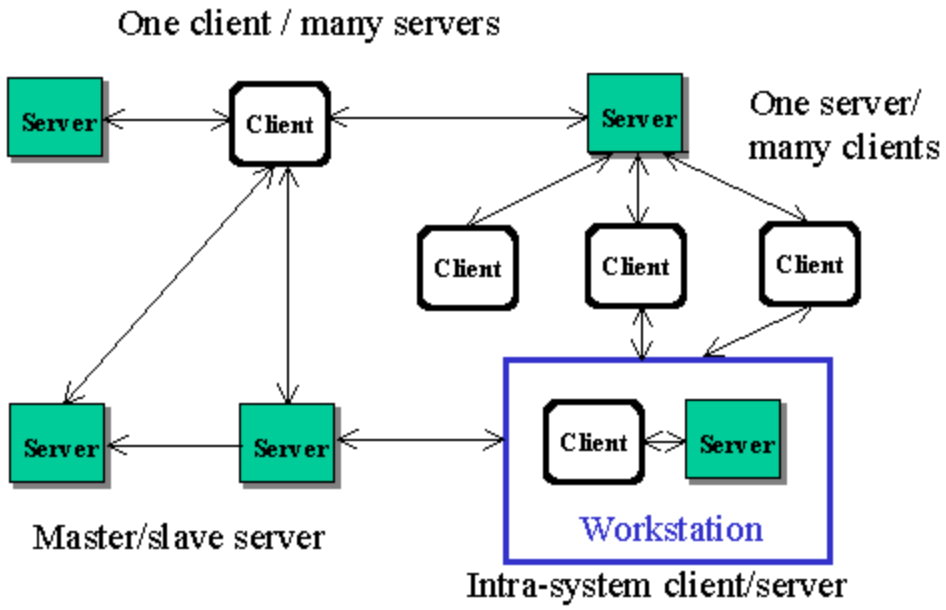
A.K. Yeung 1998-10-10 u51-09

Figure 10: Example of a GIS project workspace



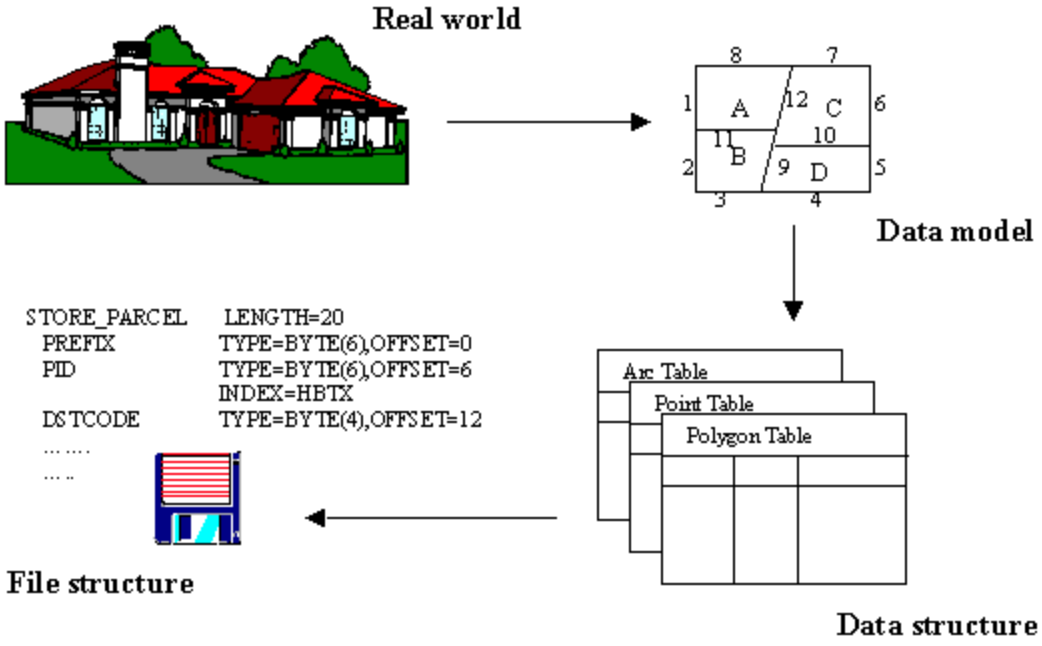
A.K. Yeung 1998-10-10 u51-10

Figure 11: Client/server computing architecture



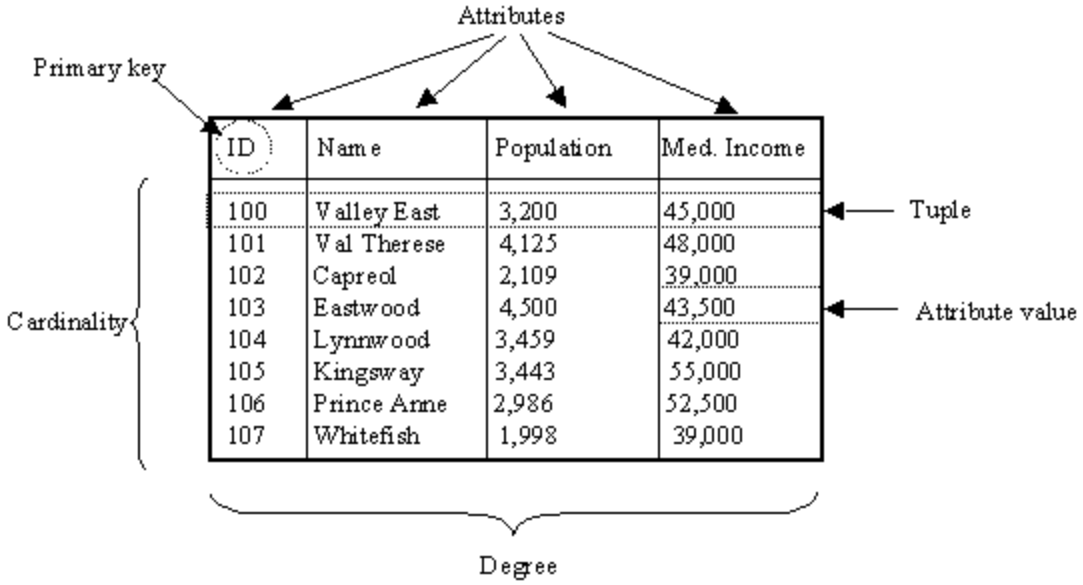
A.K. Yeung 1998-10-10 u51-11

Figure 12: Levels of abstraction in information organization



A.K. Yeung 1998-10-10u51-12

Figure 13: Characteristics of a relational table



A.K. Yeung 1998-10-10 u51-13

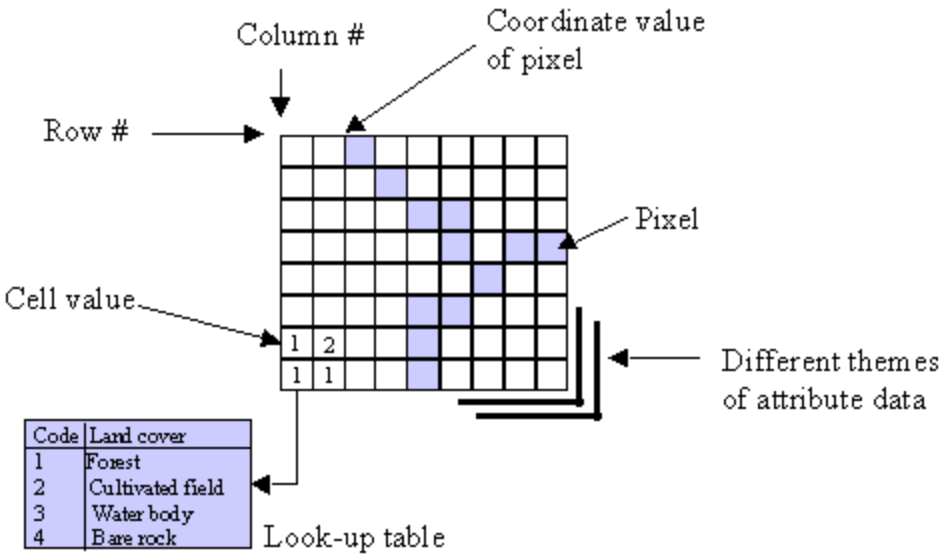
Figure 14: Defining an O-O database schema using OODDL

```
define type  Land_parcel:
    tuple(   parcel_identifier:     string,
             address_1:             string,
             address_2:             string,
             city:                  string,
             owner_first_name:      string,
             date_registered:       Date,
             assessed_value:        real,
             tax_paid:              real,
             tax_balance:           real
             .......                ........);

define type  Date:
    tuple (  year:                  integer,
             month:                 integer,
             day:                   integer);
```

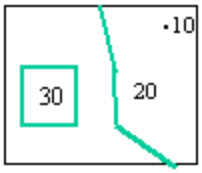
A.K. Yeung 1998-10-10u51-14

Figure 15: Characteristics of raster data structure



A.K. Yeung 1998-10-10 u51-15

Figure 16: "Spaghetti" data model and data structure



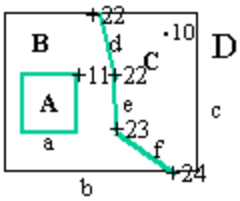
Data model

Data structure

Identifier	Coordinates
10	x,y
20	x_1,y_1 ..... x_n,y_n (string)
30	x_1,y_1 ..... x_1,y_1 (loop)

A.K. Yeung 1998-10-10 u51-16

Figure 17: Hierarchical data model and data structure



Data model

Polygon	Bounding chains
A	a
B	a,b
C	d,e,f,c

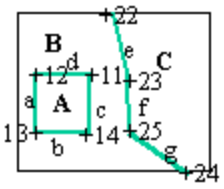
Data structure

Chain ID	From node	To node	Left poly.	Right poly.
a	11	11	A	B
b	22	24	B	--
....				
f	23	24	C	B

Node	x	y
10		
11		
...		
24		

A.K. Yeung 1998-10-10 u51-17

Figure 18: Topological data model and data structure



Data model

Data structure

Topologically coded network and polygon file

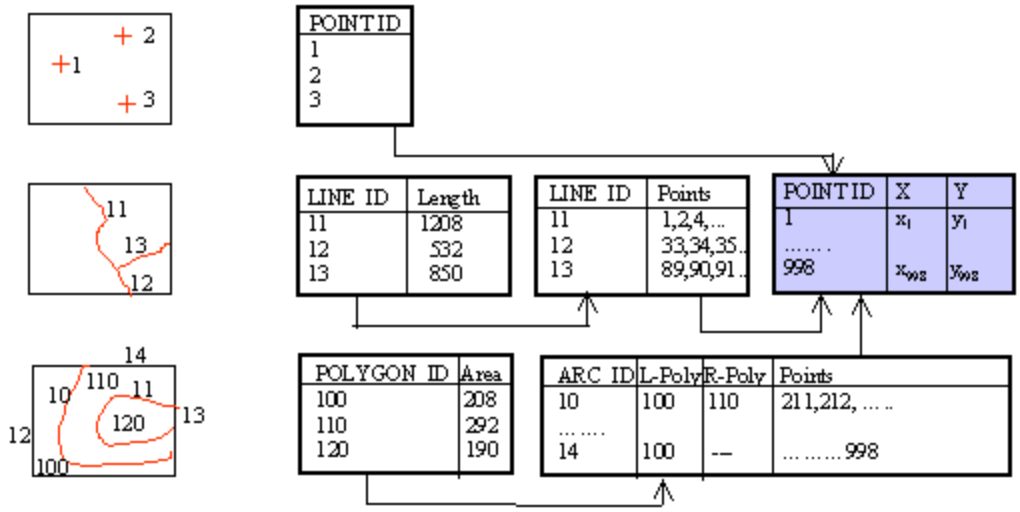
Arc ID	From node	To node	Left poly.	Right poly.
a	12	13	A	B
b	13	14	A	B
……				
…….				
f	23	25	C	B
g	25	24	C	B

X,Y coordinate node file

Node	x	y
11		
12		
…		
25		

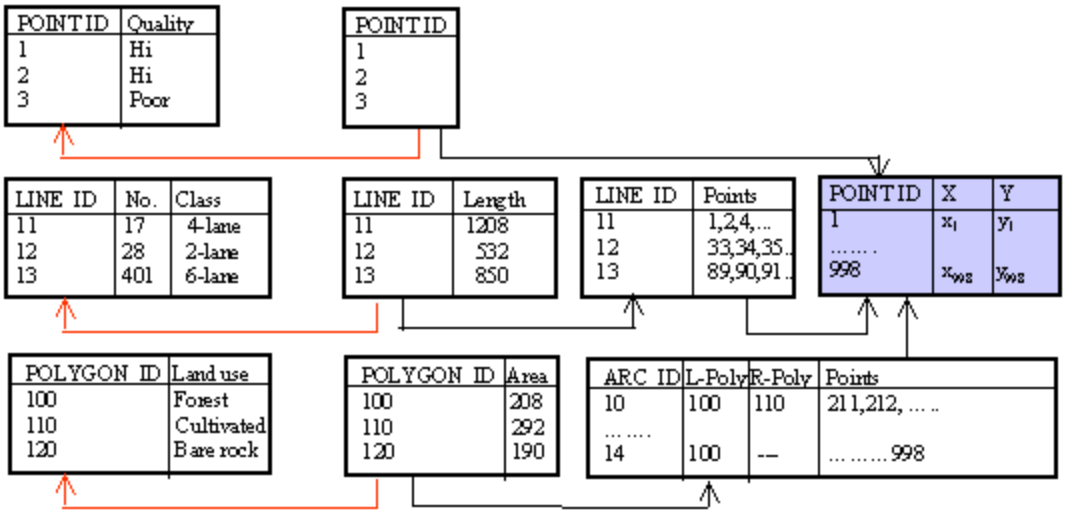
A.K. Yeung 1998-10-10 u51-18

Figure 19: The georelational data structure (1)



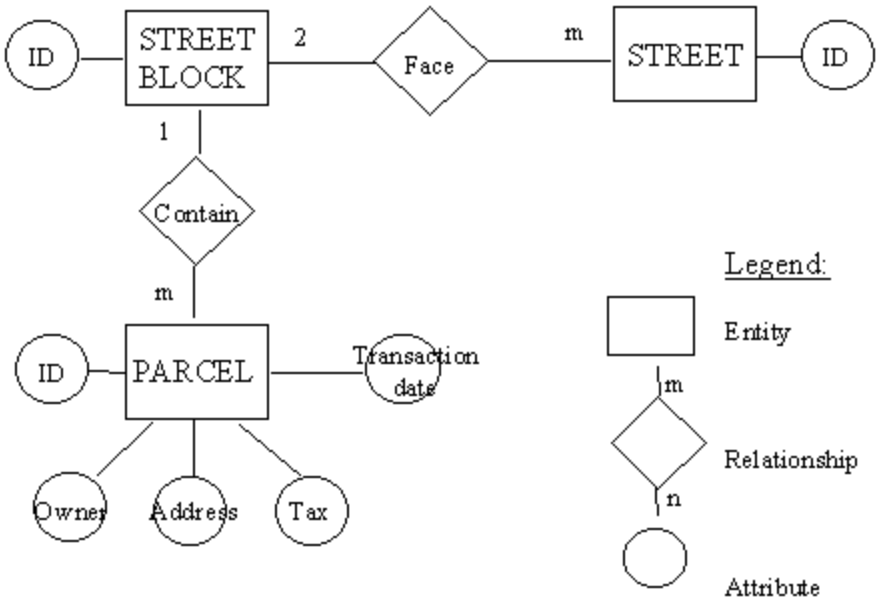
A.K. Yeung 1998-10-10 u51-19

Figure 20: The georelational data structure (2)



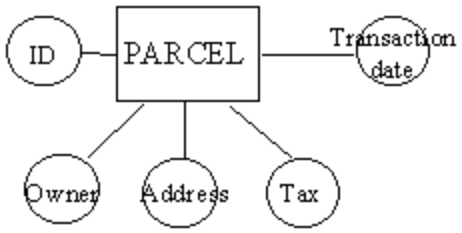
A.K. Yeung 1998-10-10 u51-20

Figure 21: A portion of an entity-relationship (E-R) diagram



A.K. Yeung 1998-10-10 u51-21

Figure 22: Logical schema of entity PARCEL in Figure 21



PARCEL (Identifier, Owner, Address, Tax, Transaction date)

Relation name

Primary key

A.K. Yeung 1998-10-10 u51-22

Figure 23: Example of a physical schema

Item Definition Table:

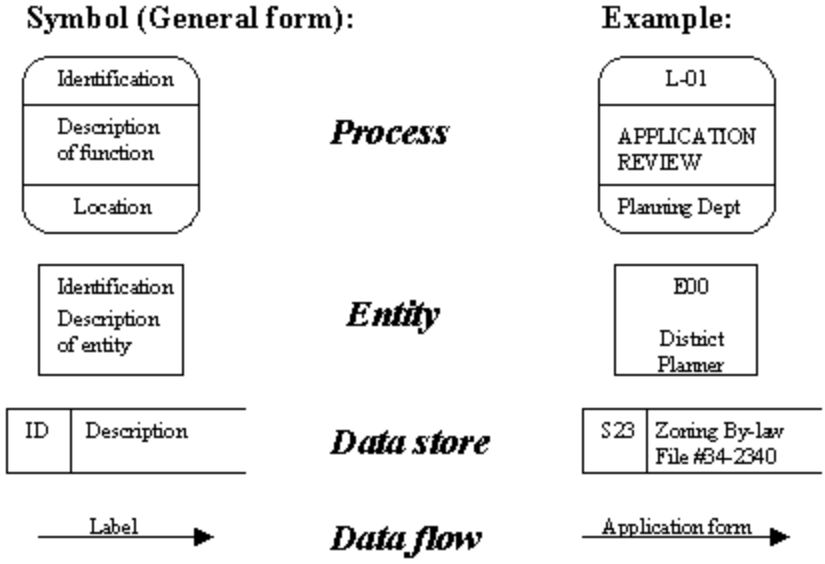
COLUMN	ITEM NAME	WIDTH	OUTPUT	TYPE	N.DEC	ALT NAME
1	PARCEL-ID	10	15	B	-	-
11	AREA	10	15	F	2	-
21	LU-CLASS	3	5	C	-	-
24	OWNER-LN	15	20	C	-	-
29	OWNER-FN	15	20	C	-	-
44	ADDRESS-1	30	35	C	-	-
74	ADDRESS-2	30	35	C	-	-
104	TRNS-DATE	8	10	B	-	-
112	ASS-VALUE	10	15	I	-	-
122	TAX-RATE	5	10	F	3	-
127	TAX	8	10	F	2	-

Explanatory notes:

PARAMETER	DESCRIPTION
Item name	Any name to 16 characters
Width	No. of space used to store item values
Output	No. of spaces used to display item values
Type	Data item type:
C	Character
I	Integer
B	Binary
N	Number
D	Date mm/dd/yr
F	Floating point
N.DEC	No. of decimal points

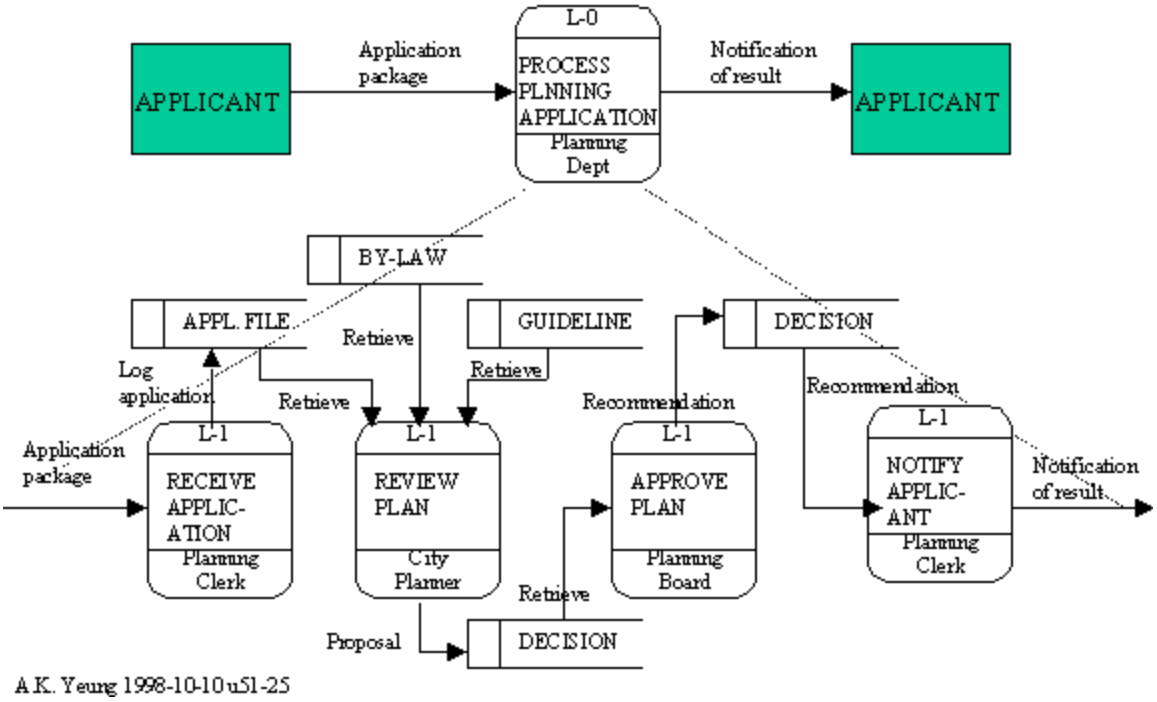
A.K. Yeung 1998-10-10 u51-23

Figure 24: Elements of a data flow diagram



A.K. Yeung 1998-10-10 u51-24

Figure 25: Processing modeling using DFDs



A.K. Yeung 1998-10-10 u51-25