Chapter 21: New Applications

- Decision-Support Systems
- Data Analysis
- Data Mining
- Data Warehousing
- Spatial and Geographic Databases
- Multimedia Databases
- Mobility and Personal Databases
- Information-Retrieval Systems
- Distributed Information Systems
- The World Wide Web

Decision-Support Systems

- Decision-support systems utilize data collected by on-line transaction-processing systems to generate summary data.
- Information for decision support can be extracted by simple SQL queries, and by SQL extensions.
- Can interface *statistical analysis* packages (S++) with databases.
- *Data mining* seeks to discover knowledge automatically, in the form of statistical rules and patterns from large databases.
- A *data warehouse* archives information gathered from multiple sources, and stores it under a unified schema, at a single site.

Data Analysis

- Aggregate functions summarize large volumes of data to support simple data analysis.
- A *histogram* partitions the values taken by an attribute into ranges, and computes an aggregate over the values in each range; cumbersome to use SQL to construct a histogram.
- *Cross-tabulation* of *number* by *size* and *color* of sample relation *sales* with the schema *Sales(color, size, number)*.

	Small	Medium	Large	Total
Light	8	35	10	53
Dark	20	10	5	35
Total	28	45	15	88

- Can represent the data in relational form by using the value **all** to represent subtotals.

Data Analysis (Cont.)

Color	Size	Number
Light	Small	8
Light	Medium	35
Light	Large	10
Light	**all**	53
Dark	Small	20
Dark	Medium	10
Dark	Large	5
Dark	**all**	35
all	Small	28
all	Medium	45
all	Large	15
all	**all**	88

- Obtain (Light, **all**, 53) and (Dark, **all**, 35) by eliminating individual tuples with different values for *size*, and by replacing the value of *number* by **sum**.

Data Analysis (Cont.)

- *Rollup*: Moving from finer-granularity data to a coarser granularity by means of aggregation.
- *Drill down*: Moving from coarser-granularity data to finer-granularity data.
- Proposed extensions to SQL, such as the **cube** operation, help to support generation of summary data.

  The following query generates the previous table.

  select *color, size,* **sum**(*number*)
  from *sales*
  groupby *color, size* **with cube**

Data Mining

- Like knowledge discovery in artificial intelligence, data mining discovers statistical rules and patterns; it differs from machine learning in that it deals with large volumes of data, stored primarily on disk.
- Knowledge discovered from a database can be represented by a set of *rules*.
- Discover rules using one of two models:

  1. The user is involved directly in the process of knowledge discovery.
  2. The system is responsible for automatically discovering knowledge from the database, by detecting patterns and correlations in the data.

Knowledge Representation Using Rules

- General form of rules: $\forall\ \overline{X} antecedent \Rightarrow consequent$

  $\overline{X}$ is a list of one or more variables with associated ranges.
- The rule $\forall\ transactions\ \ T,\ buys(T, bread) \Rightarrow buys(T, milk)$ states: if there is a tuple $(t_i, bread)$ in the relation $buys$, there must also be a tuple $(t_i, milk)$ in the relation $buys$.
- *Population*: Cross-product of the ranges of the variables in the rule.
- *Support*: Measure of what fraction of the population satisfies both the antecedent and the consequent of the rule.
- *Confidence*: Measure of how often the consequent is true when the antecedent is true.

Classes of Data-Mining Problems

- *Classification*: Finding rules that partition the given data into disjoint groups that are relevant for making a decision (e.g., which of several factors help classify a person's credit worthiness).
- Useful to determine *associations* between different items (e.g., someone who buys bread is quite likely also to buy milk).
- *Sequence correlations*: determine correlations between information and related sequenced data (e.g., When bond rates go up, stock prices go down within two days).

User-Guided Data Mining

- Primary responsibility for discovering rules is with the user, who may runs tests on the database to verify or refute a hypothesis.
- Derive from the database the confidence and support for the rule expressing the hypothesis.
- Example: Refine the hypothesis "People who hold master's degrees are the most likely to have an excellent credit rating." into the rule:

$$\forall\ people\ P,\ P.degree = Masters\ \textbf{and}\ C.income \geq 75,000$$
$$\Rightarrow \quad C.credit = excellent$$

- *Data-visualization* detect patterns in large volumes of data via maps, charts, color-coding, and other graphical representations.

Discovery of Classification Rules

- *Training set*: a data sample in which the grouping for each tuple is already known.
- Top down generation of *classification tree*.
  - Each node of the tree partitions the data into groups based on one attribute.
  - The construction of a path in the tree stops when either the attribute has properly classified the data, or all attributes have been considered.
- Credit example: Within each partition based on *degree*, the partitions defined by *income* adequately classify the tuples.

  Tree construction stops here for each partition based on *degree*.
- In general, different branches of the tree could grow to different levels.

Discovery of Association Rules

$$\forall\ transactions\ \ T,\ buys(T, bread) \Rightarrow buys(T, milk)$$

- Derive rule by associating a bitmap with each transaction, with one bit per item of interest in the shop.
- Usually desire rules with strong support, which will involve only items purchased in a significant percentage of the transactions.

$$\forall\ transactions\ T,\ buys(T, i_1)\ \textbf{and}\ \ldots\ \textbf{and}\ buys(T, i_n) \Rightarrow buys(T, i_0)$$

- Consider all subsets of the set of relevant items, and, for each set, check whether there is a sufficient number of transactions in which all the items in the set are purchased.

Discovery of Association Rules (Cont.)

- Few sets: Determine level of support via a single pass.
  - A count is maintained for each set, initially set to 0.
  - When a transaction is fetched, the count is incremented for each set of items, all of whose bits are set in the transaction's bitmap.
  - Sets with a high count at the end of the pass correspond to items with a high degree of association.
- Many sets: Cost of processing each transaction becomes correspondingly large.
  - Use multiple passes, considering only some sets in each pass.
  - Once a set is eliminated because it occurs in too small a fraction of the transactions, none of its supersets needs to be considered.

Data Warehousing

- Provides a single consolidated interface to data, making decision-support queries easier to write.
- By accessing information for decision support from a data warehouse, the decision maker ensures that on-line transaction-processing systems are not affected by the decision-support workload.
- Issues in building a warehouse:
  - When and how to gather data.
  - What schema to use.
  - How to propagate updates.
  - What data to summarize.

Spatial and Geographic Databases

- Spatial databases store information related to spatial locations and support efficient storage, indexing and querying of spatial data.
- Special purpose index structures are important for accessing spatial data, and for processing spatial join queries.
- *Design databases* (also CAD) databases store design information about how objects (e.g., buildings, aircraft) are constructed, or layouts of integrated-circuits.
- *Geographic databases* store geographic information (e.g., maps); often called *geographic information systems* or GIS.

Representation of Geometric Information

- Various geometric constructs can be represented in a database in a normalized fashion.
- Represent a line segment by the coordinates of its endpoints.
- Approximate a curve by partitioning it into a sequence of segments; represent each segment as a separate tuple that also carries with it the identifier of the curve (2D features such as roads).
- Closed polygons: list its vertices in order, starting vertex is the same as the ending vertex.

  Alternative: Triangulation — give polygon an identifier; each of the triangles into which it is divided carries the identifier.

Representation of Geometric Information (Cont.)

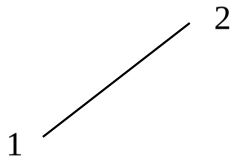
- Representation of points and line segments in 3-D similar to 2-D, except that points have an extra z component.

- Represent arbitrary polyhedra by dividing them into tetrahedrons, like triangulating polygons.

  Alternative: List their faces, each of which is a polygon, along with an indication of which side of the face is inside the polyhedron.

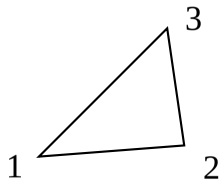
Representation of Geometric Constructs

line segment



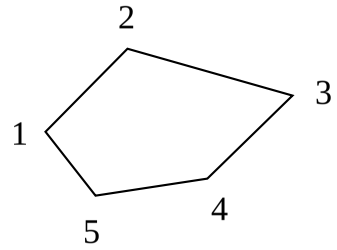
{(x1,y1), (x2,y2)}

triangle



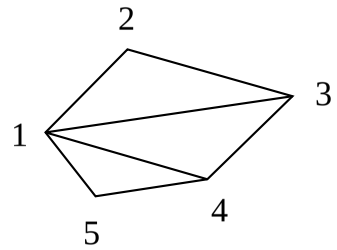
{(x1,y1), (x2,y2), (x3,y3)}

polygon



{(x1,y1), (x2,y2), (x3,y3), (x4,y4), (x5,y5)}

polygon



{(x1,y1), (x2,y2), (x3,y3), ID1}
{(x1,y1), (x3,y3), (x4,y4), ID1}
{(x1,y1), (x4,y4), (x5,y5), ID1}

object **representation**

Design Databases

- Represent design components as objects (generally geometric objects); the connections between the objects indicate how the design is structured.
- Simple two-dimensional objects: points, lines, triangles, rectangles, polygons.
- Complex two-dimensional objects: formed from simple objects via union, intersection, and difference operations.
- Complex three-dimensional objects: formed from simpler objects such as spheres, cylinders, and cuboids, by union, intersection, and difference operations,
- *Wireframe* models represent three-dimensional surfaces as a set of simpler objects.

Geographic Data

- Raster data consist of bit maps or pixel maps, in two or more dimensions.
  - Example 2-D raster image: satellite image of cloud cover, where each pixel stores the cloud visibility in a particular area.
  - Additional dimensions might include the temperature at different altitudes at different regions, or measurements taken at different points in time.
- Design databases generally do not store raster data.

Geographic Data (Cont.)

- Vector data are constructed from basic geometric objects: points, line segments, triangles, and other polygons in two dimensions, and cylinders, spheres, cuboids, and other polyhedrons in three dimensions.
- Vector format often used to represent map data.
  - Roads can be considered as two-dimensional and represented by lines and curves.
  - Some features, such as rivers, may be represented either as complex curves or as complex polygons, depending on whether their width is relevant.
  - Features such as regions and lakes can be depicted as polygons.

Geographic Data (Cont.)

- *Geographic Information Systems* (GIS), provide location information.
- Vehicle navigation systems store information about roads and services for the use of drivers.
- *Global Positioning System* or *GPS* unit – utilizes information broadcast from GPS satellites to find the current location with an accuracy of tens of meters.

Spatial Queries

- Nearness queries request objects that lie near a specified location.
- Region queries deal with spatial regions, e.g., ask for objects that lie partially or fully inside a specified region.
- Queries that compute intersections of regions – *spatial join* of two spatial relations with the location playing the role of join attribute.

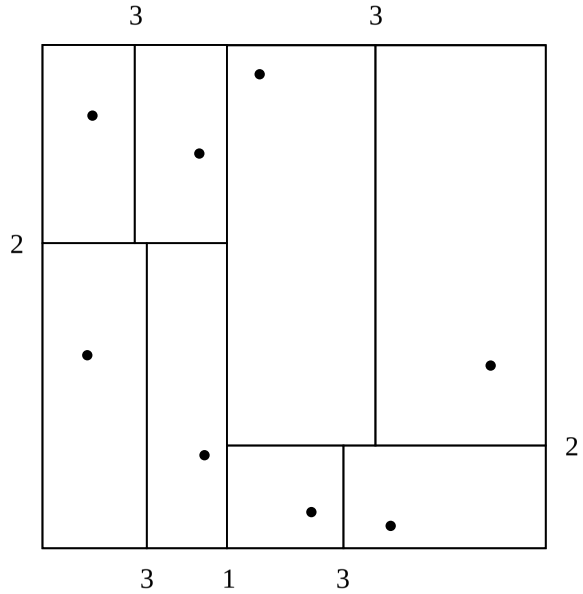
Spatial Queries (Cont.)

- Spatial data is typically queried using a graphical query language; results are also displayed in a graphical manner.
- Graphical interface constitutes the front-end; extensions of SQL with abstract data types, such as lines, polygons and bit maps, have been proposed as a back-end.
  - allows relational databases to store and retrieve spatial information
  - queries can mix spatial and nonspatial conditions
  - extensions also include and allowing spatial conditions (*contains* or *overlaps*).

Indexing of Spatial Data

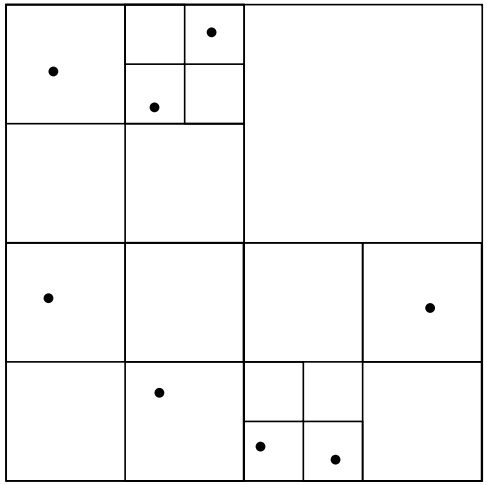
- *k-d* tree – early structure used for indexing in multiple dimensions.
- Each level of a *k-d* tree partitions the space into two.
  - one dimension – at the node at the top level of the tree.
  - other dimension – in nodes at the next level and so on, cycling through the dimensions.
- In each node, approximately half of the points stored in the sub-tree fall on one side and half on the other.
- Partitioning stops when a node has less than a given maximum number of points.
- The *k-d-B tree* extends the k-d tree to allow multiple child nodes for each internal node; well-suited for secondary storage.

Division of Space by a k-d Tree



- Each line corresponds to a node in the tree, and the maximum number of points in a leaf node has been set at 1.
- Each line in the figure (other than the outside box) corresponds to a node in the k-d tree.
- The numbering of the lines in the figure indicates the level of the tree at which the corresponding node appears.

Quadtrees



- Each node is associated with a rectangular region of space; the top node is associated with the entire target space.
- Each non-leaf node divides its region into four equal sized quadrants, and correspondingly each such node has four child nodes corresponding to the four quadrants.
- Leaf nodes have between zero and some fixed maximum number of points (set to 1 in example).

Quadtrees (Cont.)

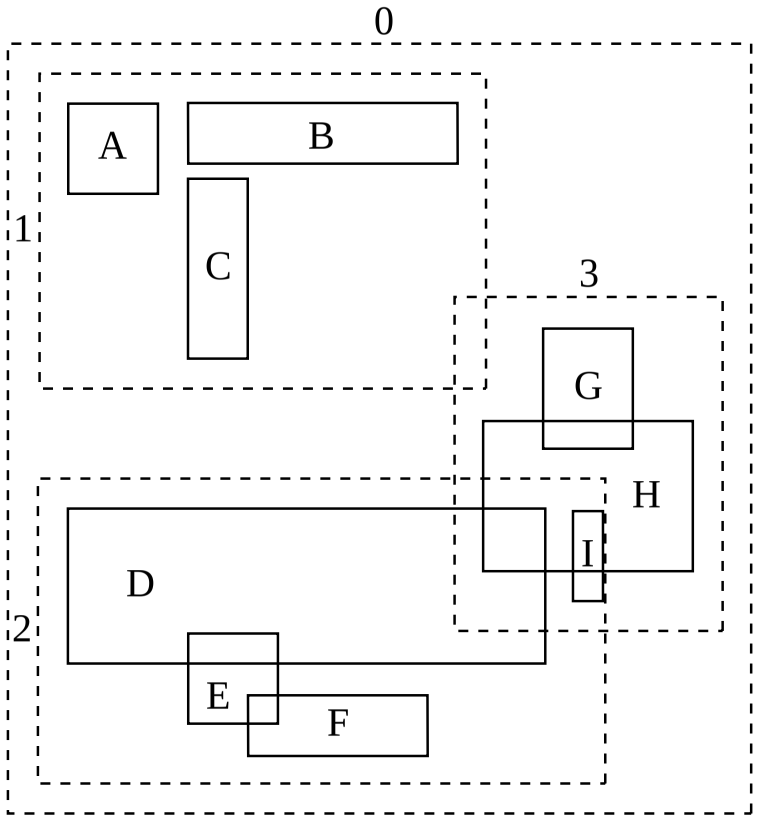
- *PR quadtree*: stores points; space is divided based on regions, rather than on the actual set of points stored
- *Region quadtrees* store array (raster) information.
  - A node is a leaf node if all the array values in the region that it covers are the same. Otherwise, it is subdivided further into four children of equal area, and is therefore an internal node.
  - Each node corresponds to a subarray of values.
  - The subarrays corresponding to leaves either contain just a single array element, or have multiple array elements, all of which have the same value.
- Extensions of k-d trees and quadtrees handle indexing of line segments and polygons.

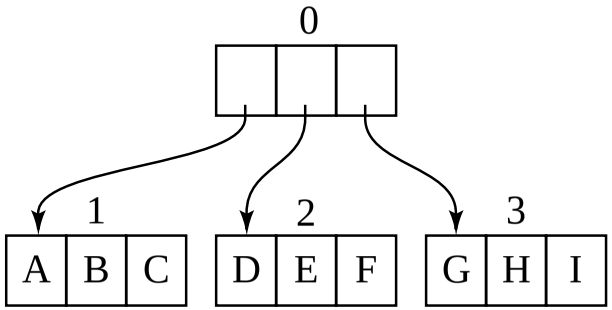
R-Trees

- R-trees are a two-dimensional extension of B-trees and, with variants like R+-trees and R*-trees, are useful for indexing rectangles and other polygons.
- A rectangular *bounding box* is associated with each tree node.
- The bounding box associated with a non-leaf node contains the bounding box associated with all its children.
- A polygon is stored only in one node, and the bounding box of the node must contain the polygon.
- The storage efficiency or R-trees is better than that of k-d trees or quadtrees since a polygon is stored only once.

Example R-Tree

A set of rectangles (solid line) and the bounding boxes (dashed line) of the nodes of an R-tree for the set of rectangles.
The R-tree is shown on the right.





Multimedia Databases

- To provide such database functions as indexing and consistency, it is desirable to store multimedia data in a database (rather than storing them outside the database, in a file system).
- The database must handle large object representation.
- Similarity-based retrieval must be provided by special index structures.
- Must provide guaranteed steady retrieval rates for *continuous-media data*.

Similarity-Based Retrieval

- Pictorial data — Two pictures or images that are slightly different as represented in the database may be considered the same by a user (identify similar designs for registering a new trademark).
- Audio data — Speech-based user interfaces allow the user to give a command or identify a data item by speaking (test user input against stored commands).
- Handwritten data — Identify a handwritten data item or command stored in the database (requires similarity testing).

Continuous-Media Data

- Most important types are video and audio data.
- Characterized by high data volumes and real-time information-delivery requirements:
  - Data must be delivered sufficiently fast that no gaps in the audio or video result.
  - Data must be delivered at a rate that does not cause overflow of system buffers.
  - Synchronization among distinct data streams must be maintained (video of a person speaking must show lips moving synchronously with the audio).

Multimedia Data Formats

- Store and transmit multimedia data in compressed form; JPEG is the most widely used format for image data.
- Encoding each frame of a video using JPEG is wasteful, since successive frames of a video are often nearly the same.
- MPEG standards use commonalities among a sequence of frames to achieve a greater degree of compression.
- MPEG-1 stores a minute of 30-frame-per-second video and audio in approximately 12.5 MB (compares with 75 MB for video using only JPEG); quality comparable to VHS video tape.
- MPEG-2 designed for digital broadcast systems and digital video disks; negligible loss of video quality.

  Compresses 1 minute of audio-video to approximately 2.25 MB.

Video Servers

- Current video-on-demand servers are based on file systems; existing database systems do not met real-time response requirements.
- Video server — Multimedia data are stored on several disks (RAID configuration), or on tertiary storage for less frequently accessed data.
- Head-end terminals — used to view multimedia data; PCs or TVs attached to a small, inexpensive computer called a *set-top box*.
- Network — Transmission of multimedia data from a server to multiple head-end terminals requires a high-capacity network; often use asynchronous-transfer-mode (ATM) networks.

Mobility and Personal Databases

- The mobile computing environment consists of mobile computers, referred to as *mobile hosts*, and a wired network of computers.
- Mobile hosts communicate to the wired network via computers referred to as *mobile support stations*.
- Each mobile support station manages those mobile hosts within its *cell*.
- Mobile hosts may move between cells, thus necessitating a *handoff* of control from one mobile support station to another.
- Without fixed locations and network addressed, can become difficult to determine the optimal location to materialize a query result; user's locations may be a parameter of the query.

Routing and Query Processing

Must consider these competing costs:

- User time.
- Connection time — used to assign monetary charges in some cellular systems.
- Number of bytes, or packets, transferred — used to compute charges in digital cellular systems
- Time-of-day based charges — vary based on peak or off-peak periods
- Energy — optimize use of battery power; different forms of data transfer present different power demands (requires less energy to receive than to transmit radio signals).

Broadcast Data

- Mobile support stations can broadcast frequently-requested data; allows mobile hosts to wait for needed data, rather than having to consume energy transmitting a request.
- A mobile host may optimize energy costs by determining if a query can be answered using only cached data; if not then must either:
  - Wait for the data to be broadcast
  - Transmit a request for data and must know when the relevant data will be broadcast.
- Broadcast data may be transmitted according to a fixed schedule or a changeable schedule.
  - For changeable schedule – the broadcast schedule must itself be broadcast at a well-known radio frequency and at well-known time intervals.

Disconnectivity and Consistency

- A mobile host may remain in operation during periods of disconnection.
- Problems created if the user of the mobile host issues queries and updates on data that resides or is cached locally:
  - **Recoverability:** Updates entered on a disconnected machine may be lost if the mobile host fails. Since the mobile host represents a single point of failure, stable storage cannot be simulated well.
  - **Consistency:** Cached data may become out of date, but the mobile host cannot discover this until it is reconnected.

Mobile Updates

- Partitioning via disconnection is the normal mode of operation in mobile computing.
- For data updated by only the mobile host, simple to propagate update when mobile host reconnects; in other cases data may become invalid and updates may conflict.
- When data are updated by other computers, *invalidation reports* inform a reconnected mobile host of out-of-date cache entries; however, mobile host may miss a report.
- Version-numbering–based schemes guarantee only that if two hosts independently update the same version of a document, the clash will be detected eventually, when the hosts exchange information either directly or through a common host.
- Facilitating the reconciliation of inconsistent copies of data still under research.

Information Retrieval Systems

- Information retrieval systems use a simpler data model than database systems, but provide more powerful querying capabilities within the restricted model.
- Queries attempt to locate documents that are of interest by specifying, for example, sets of keywords.
- The query a user has in mind usually cannot be stated very precisely, and hence information retrieval systems order answers based on their potential relevance.
- *Similarity based retrieval* – similarity may be defined based on keywords or other metrics (i.e., similarity of images in an image database).

Indexing of Documents

- Documents that contain a specified keyword can be located using an *inverted index*, which maps each keyword K_i to the set S_i of documents that contain K_i.
- Storing the index for approximate retrieval saves space.
  - *false drop* – a few relevant documents may not be retrieved.
  - *false positive* – a few irrelevant documents may be retrieved.
  - Index should not permit *any* false drops, but may permit a few false positives.
- Relevant performance metrics:
  - Precision — what percentage of the retrieved documents are relevant to the query.
  - Recall — what percentage of the documents relevant to the query were retrieved.

Indexing of Documents (Cont.)

- *and* operation — Finds documents that contain all of a set of keywords $K_1, K_2, \ldots, K_n$. Retrieve the corresponding sets of documents $S_1, S_2, \ldots, S_n$. The *intersection*, $S_1 \cap S_2 \cap \ldots \cap S_n$, constitutes the desired set of documents.
- *or* operation — gives the set of all documents that contain at least one of the keywords $K_1, K_2, \ldots, K_n$ by computing the union, $S_1 \cup S_2 \cup \ldots \cup S_n$, of the sets.
- *but not* operation — finds documents that do not contain a specified keyword K_i. Given a set of document identifiers S, we can eliminate documents that contain the specified keyword K_i by taking the difference $S - S_i$, where S_i is the set of identifiers of documents that contain the keyword K_i.

Browsing and Hypertext

- *Hypertext* systems take the idea of storing document identifiers or pointers and provide a facility where the user can easily switch from one document to another.
- Typically use a point-and-click interface, where a simple mouse click on the display screen on top of the referred document retrieves and displays it.
- *Hypermedia* systems provide not only text, but also other media such as images, videos, and audio clips.
- *Distributed hypertext* systems permit references to documents stored at other sites in a distributed system such as the World Wide Web.

Distributed Information Systems

- Distributed information systems running on the Internet have seen explosive growth in recent years.

- The World Wide Web system supports browsing such information using the hypertext paradigm.

- Automated tools for locating and indexing information in such distributed heterogeneous systems have also been developed; provide standardized ways of accessing data and standardized GUIs.

The Gopher and WAIS

- A Gopher system consists of servers and clients.
  - Server organizes data into directories.
  - Client initially communicates with a server; the top level directory of the server hierarchy is displayed as a menu.
  - Menu item can be another directory in the hierarchy, a document, or a link to a directory on another server.
  - Allows a seamless connection to remote servers.
- Information retrieval in Gopher is based on browsing and navigating a directory hierarchy.
- Wide Area Information System (WAIS), retrieves information via a powerful keyword based indexing mechanism.
- Each site maintains a *site description*; describes the kind of information stored at other sites, and how to access them.

The World Wide Web

- A distributed information system based on hypertext.
- Most Web documents are hypertext documents formatted via the HyperText Markup Language (HTML), which is based on the Standard Generalized Markup Language (SGML).
- HTML documents contain text along with font specifications, and other formatting instructions; links to other documents can be associated with regions of the text.
- The displayed document is a hypertext document; with an appropriate browser, the user can click on a region that has a link associated with it, and the document pointed to by the link is then displayed.
- The programming language *Java* allows documents to contain programs that are executed at the user's site; thus, documents can be *active*, rather than just passive.

Universal Resource Locators

- In the Web, the functionality of pointers is provided by *Universal Resource Locator*s (URLs).
- URL example: http://www.bell-labs.com/foo/bar
  - The first part indicates how the document is to be accessed; "http" indicates that the document is to be accessed using the HyperText Transfer Protocol.
  - The second part gives the unique name of a machine on the Internet.
  - The rest of the URL is the path name of the file on the machine.

Universal Resource Locators (Cont.)

- With Gopher, what is displayed is either a document or a directory; a HTML document display is simultaneously a document and a directory.
- Gopher systems are set up and controlled by system administrators; anyone connected to the Internet can create documents on the Web, and give the URL to anyone else.
- *Home pages* – contain information about users and their work.

Web Servers

- A Web server can easily serve as a front end to a variety of information services.
- The document name in a URL may identify an executable program, that, when run, generates a HTML document.
- When a HTTP server receives a request for such a document, it executes the program, and sends back the HTML document that is generated.
- The Web client can pass extra arguments with the name of the document.
- To install a new service on the Web, one simply needs to create and install an executable that provides that service.
- The HTML language supported by the Web provides a graphical user interface to the information service.

Display Languages

- Text markup languages such as the Standard Generalized Markup Language (SGML) fill a void between plain text and page description/text formatting languages.

- SGML provides a grammar for specifying document formats based on standard markup annotations.

- In addition to formatting, hypertext link, and image display commands, HTML provides some limited input features.

Java

- Java language allows documents to be *active* (e.g., animation by executing programs at the local site).
- Java programs can be stored at server sites (like HTML documents), and can be downloaded and executed by any client site.
- Benefits of Java:
  - Flexible interaction with the user.
  - Executing programs at the client site speeds up interaction greatly, compared to every interaction being sent to a server site for processing.
- Java's security system ensures that the Java code does not make any system calls directly. It notifies the user about potentially dangerous actions, such as file writes, and allows the option to abort the program or to continue execution.

Web Interfaces to Databases

- Extremely useful to link databases used for transaction processing with the Web.

  Example: Information filled in on an HTML order form can be executed as a database transaction; the results can be formatted into HTML and displayed to the user.

- Fixed HTML sources for display to users have limitations:
  - Cannot customize fixed Web documents for individual users.
  - Problematic to update Web documents, especially if multiple Web documents replicate data.

Web Interfaces to Databases (Cont.)

- Generate Web documents dynamically — A document request executes a program to run queries on the database and to generate a document based on the results.
- Tailor the display based on user information stored in the database.
- Define data in Web documents by queries on a database; whenever relevant data in the database are updated, the documents will be updated too.

Web Interfaces to Databases (Cont.)

- Web interfaces to databases simplify the tasks of format conversions from HTML to SQL and from database results into HTML.
- Define a HTML document in a macro language with embedded SQL queries.

  Variables defined in HTML forms can be used directly in the embedded SQL queries.
- When the document is requested, a macro processor executes the SQL queries, and generates the actual HTML document that is sent to the user.

Locating of Information on the Web

- The Archie system automatically follows Gopher links to locate information, and creates a centralized index of information found from various sites.
- *Web crawlers* follow the hypertext links in documents to find other documents, and build an index on the documents.
- These systems run a background process to
  - find new sites.
  - obtain updated information from known sites.
  - discard defunct sites.
- Since no central authority is required for registering documents, users can easily create new documents and locate the documents via an index.