UNIT 11 DATABASE SYSTEMS

Structure

11.0 OBJECTIVES

After reading this Unit, you will be able to :

- understand various types of database systems and their applications;
- comprehend the concepts involved in the emerging data mining and data warehousing techniques; and
- understand the basics of artificial intelligence and the key areas of its applications.

11.1 INTRODUCTION

Database systems have permeated all spheres of life. This Unit dwells on some of the database systems with specific application areas. The concept of distributed database systems, has gained importance, as the location of data need not be in a single centralized place. The reason for storing the data in different locations are many. Management Information Systems (MIS) for different levels of management basically provide information support.

In the distributed data processing environment, the concept of a transaction (which is a programme) is an important area to be understood by the database producers. Data warehousing is an analytical database designed to be of help to users for making better decisions. Data warehouse is the foundation of Decision Support System (DSS). The concepts of data warehousing and data mining are interrelated. When a data warehouse is created, data mining is used to search hidden part of data. Data mining is basically using some automated tools in this process.

The technologies that attempt to develop machines to realise human like qualities such as learning, reasoning, communicating etc. are covered under Artificial

Intelligence (AI). Expert system is a closely related term, which is basically a set of AI programmes which may represent human expertise in a specific domain. All these concepts have been explained in detail.

11.2 DISTRIBUTED DATABASE SYSTEMS

A distributed database is a single logical database which is fragmented and the fragments spread across computers at different locations that are interlinked by a data communication network to provide integrated access to the data. A distributed database environment requires the data to be shared. A distributed database gives geographical data independence, i.e., a user requesting for data need not know at which site the data is located. This property is often referred to as location transparency and each local site is called a node.

11.2.1 Architecture of Distributed Databases

The architecture and schematic representation of a distributed database is shown in Fig.4.1

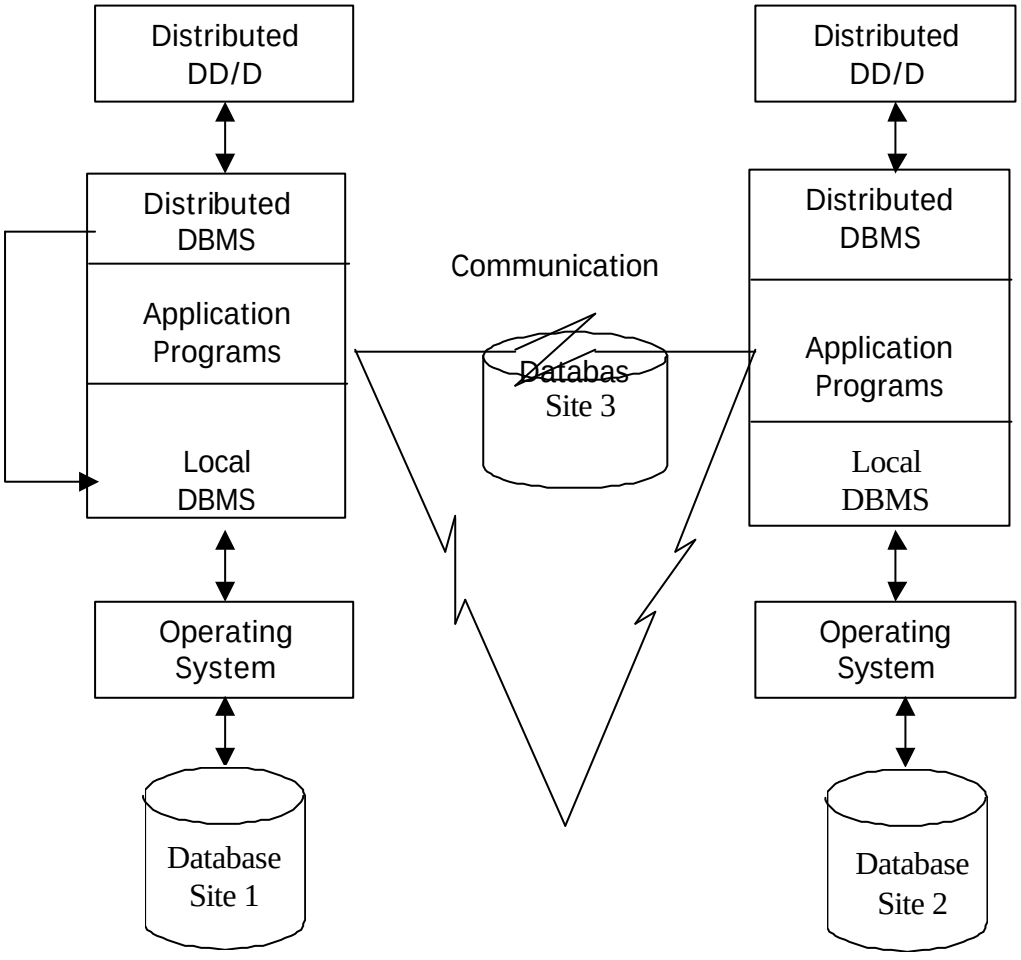

Fig: 4.1 : Architecture and schematic representation of a distributed database

As is clear from Fig.4.1, each site has a local DBMS as well as a copy of the distributed DBMS. Distributed data dictionary/directory (DD/D) stores information on the location of data in the network as well as data definitions. A request for data is first checked from the distributed data dictionary/directory for location of the required data. In case the data is available at the local site, the distributed DBMS

forwards the request to the local DBMS for processing. If the request involves data from other sites, the distributed DBMS routes the request to these sites.

When different nodes in a distributed database have mixed DBMSs (i.e., node 1 may have relational DBMS and node 2 network DBMS) then this distributed DBMS which is capable of handling such an environment is called heterogeneous distributed database management system.

Distributed database management exploits all advantages of centralized and decentralized processing. A decentralized database like a distributed database is also stored on different computers at multiple locations but in this case the computers are not interconnected and hence the data cannot be shared.

11.2.2 Justifications and Options for Distributing Data

The justifications for distributing data can be summed up as follows:

— A distributed database provides increased reliability and availability. Compared to a centralized system, which on failure becomes unavailable to all users, a distributed system will continue to function, though at a reduced level, even when a node fails.

— By encouraging local control of data at different sites, data integrity improves and data administration becomes easier.

— Distribution of data can improve access time if local data is stored locally. By locating data closer to the point of its use, communication cost can be reduced and query response time improved.

— A distributed system facilitates modular growth. New nodes hosting additional database fragments can be added to the system.

There are a number of options available for distributing data in a distributed database. These options include: i) data replication, ii) horizontal partitioning, iii) vertical partitioning, and iv) a combination of the above.

In case of **data replication** a copy of the database is stored at a few or all sites (full replication). Reliability, saving in telecommunication charges and faster response are the advantages of this option. But additional storage requirements and difficulty in propagating updates are the basic drawbacks. This option is suitable in case updates are infrequent and database interaction is restricted to read-only. CD-ROM (compact disk read only memory) offers an excellent medium for replicated databases.

Horizontal partitioning of a database involves distributing rows of a relation to multiple sites. New relations (partitions) with the requisite rows are created for this purpose. The original relation can be reconstructed by taking the union of the new relations. Horizontal partitioning can optimize performance by storing fragments of the database at the sites where they are most used.

On **vertical partitioning** of a database, selected columns of a relation are projected into new relations which are stored at different sites. The main criterion for vertical partitioning is specific data item requirements at individual sites.

A combination of the mentioned options of data distribution may be used depending upon the needs of the distributed system. The basic principle which one must keep in mind is that data should be stored at sites where it will be most frequently used.

11.3 DATABASE SYSTEMS FOR MANAGEMENT SUPPORT

Database systems for management support are broadly referred to as Management Information Systems (MIS). However, for different levels of management, database systems have been categorised based on management functions and expected outputs. Fig.4.2 illustrates a hierarchy of information systems corresponding to the three levels of management.

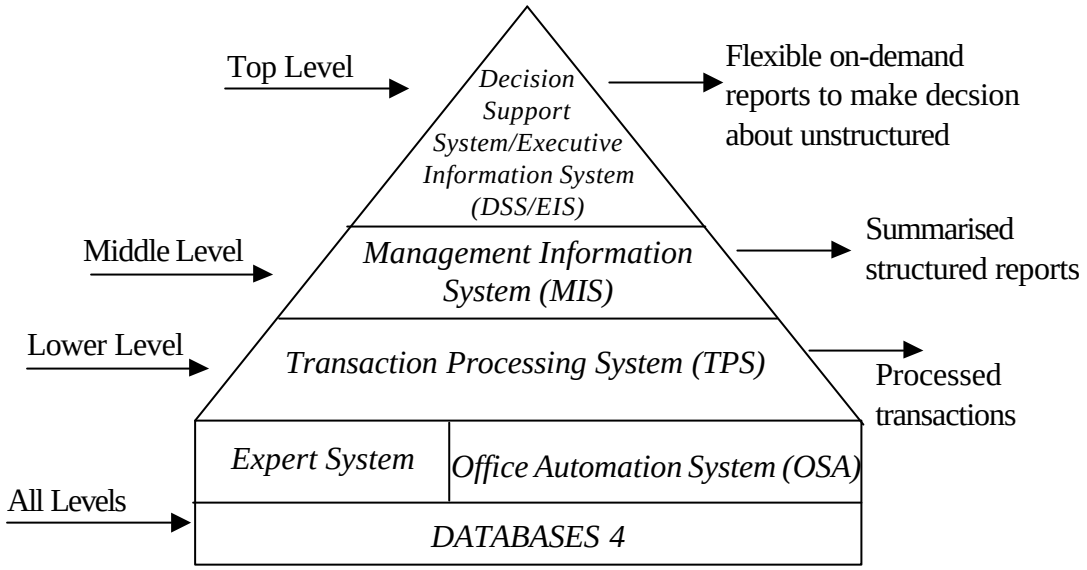

Fig. 4.2 : Information systems and management levels

It should be clear from the figure that, for lower level of managers, transaction-processing systems (TPS) yielding processed transactions (bills, orders etc) suffice. Middle level managers need management information systems providing summarised structured reports. At the top level, decision support systems (DSS) or executive information systems (EIS) capable of providing brief on-demand reports about unstructured queries are required. Office automation systems and expert systems are used by all levels including non-management.

Self Check Exercise

1) What are the advantages of distribute databases?

Note: i) Write your answer in the space given below.

ii) Check your answer with the answers given at the end of this unit.

.....

.....

.....

.....

.....

.....

11.4 TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL

Transaction management and concurrency control are the issues which play an important role in maintaining the quality of data in a distributed processing environment.

A transaction is a logical unit of operation on a database which may specify a query or addition, deletion or modification of records. A transaction cannot have a partial effect on the database, i.e., either it is completed or, if aborted, any changes to the database which the transaction made before aborting are removed. When a transaction is completed successfully and the changes brought about by the transaction made in the database, we say that the transaction has been committed.

A transaction which represents a programme is made up of three basic parts viz., begin transaction, statements pertaining to operation on data and end transaction followed by commit. A sample transaction programme has been shown in Fig.4.3.

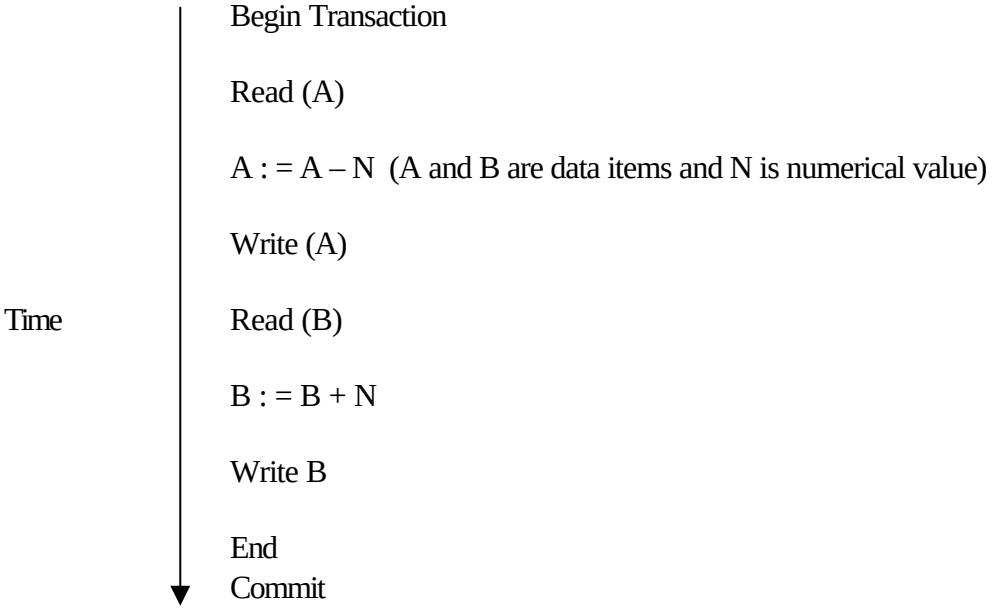

Fig. 4.3 : A simple transaction

The transaction shown in the figure is a small transaction which reads data items A and B and updates them.

When two or more transactions require the same data items for their operation, i.e., same data is shared by different transactions running together we have concurrency of access. This concurrency can create problems in the consistency of a database i.e. the database may not give the right answer to the queries. The set of rules for handling transactions in a way that the database remains consistent is called transaction management. Transactions can be executed serially or concurrently. In case of serial execution the transactions are executed one after another, i.e, the transaction T2 begins only after the transaction T1 is complete and so on (Fig. 4.4). On serial execution, there is no interference and database consistency is preserved. But at the same time there is no shareability of data.

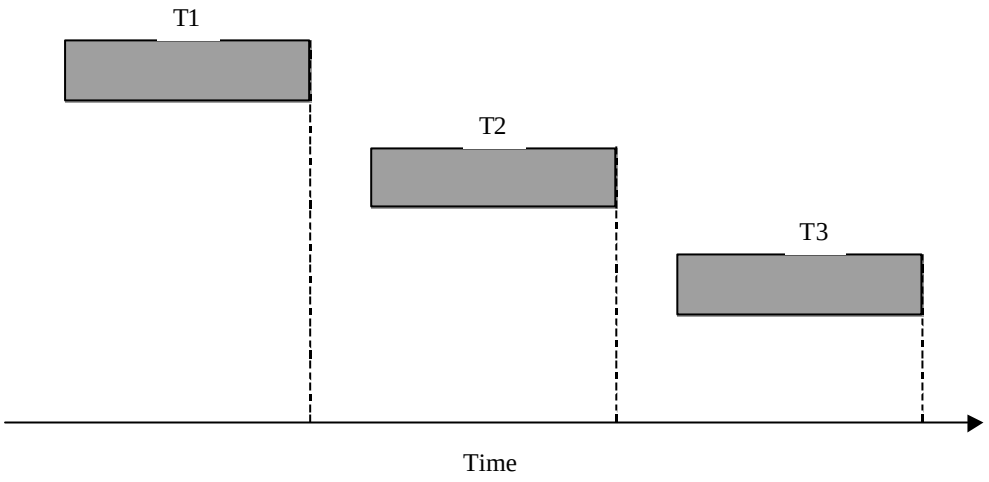

Fig. 4.4 : Serial execution of transactions

Concurrent execution promotes shareability of data. However, the problems of lost updates (i.e., when an update by one transaction is destroyed by another) and dirty reads (i.e. reading values which are not part of the database) have to be taken care of in this mode of execution. Concurrent execution has been illustrated in Fig.4.5.

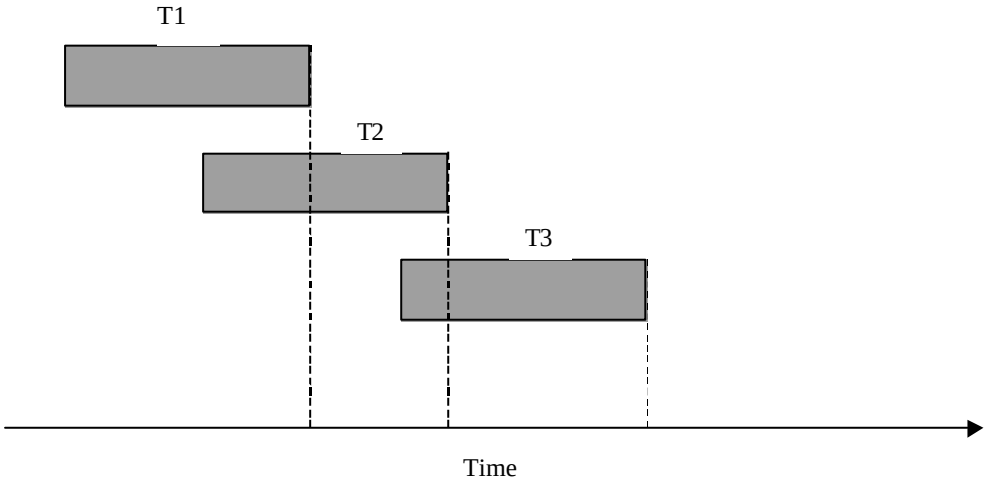

Fig. 4.5 : Concurrent execution of transactions

It should be noted that concurrent transactions which require only read operation do not cause any consistency problems. The difficulty arises only when some of the transactions have also write operation in their programme.

Consistency problems in concurrent execution are resolved by serialization. Serialization is the process of managing transactions in such a manner that their concurrent execution produces the same end result as in serial execution.

Serialization of transaction is implemented by two operations, viz., locks and releases (unlocks). Locks can be shared or exclusive. A shared lock which is also called a read lock is applied in case of read operations. A data item can be read by multiple users simultaneously under share lock mode. If a transaction T1 has a shared lock on data item A, transaction T2 can also apply a shared lock on A.

An exclusive lock or write lock is used when there are write operations or read and write operations in concurrent transactions. A data item can be modified only under exclusive lock. If a transaction T1 has exclusive lock on data item A, transaction T2 cannot apply any lock on A.

If a transaction locks a data item and is waiting for a data item locked by another transaction which is also waiting for the data item locked by the first, we have a case of deadlock (also known as deadly embrace). The only way to resolve such a situation is to abort one of the transactions. When a transaction fails the process of undoing the changes it made in the data items, it is called rollback. If several transactions are running concurrently and the first transaction which modifies data items to be used by subsequent transactions fails and the changes rolled back, all the other transactions will also fail, leading to what is known as cascade rollback.

In the running of a transaction three phases can be identified: locking, execution (data modification) and unlocking. A number of options are available in locking and unlocking phases. One such widely known option called incremental locking and simultaneous release has been illustrated in Fig.4.6. Here the transaction locks data items incrementally, i.e., only when required. The transaction locks a data item, then modifies it (execution phase) and locks the next data item for execution and so on. If any of the requested data item is not free, i.e., it is locked by another transaction, then the transaction waits for the item to become free.

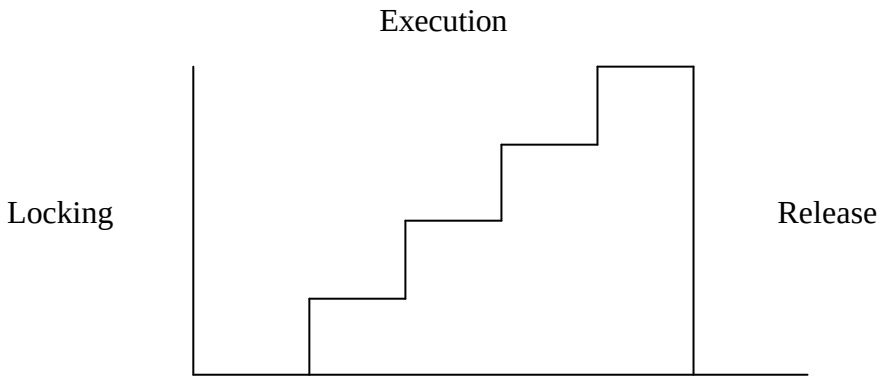

Fig. 4.6 : Three phases of a transaction. Incremental locking and simultaneous release

11.5 DATA WAREHOUSING AND DATA MINING

A data warehouse is an analytical database that is used as the foundation of a decision support system. Unlike operational database systems which are designed for day-to-day business activities, a data warehouse is an analytical system designed to help users make better decisions by analysing summarised snapshots of corporate performance.

The distinction between online transaction processing (OLTP) and online analytical processing (OLAP) should be noted clearly. OLTP involves real time transactions in operational systems which are generally very large databases (VLDB) with high throughput of transaction data. OLAP, on the other hand, focuses on retrieving data usually through detailed analysis and complex queries typically for decision support activities. Complex queries in operational systems impact on their performance and hence are not recommended.

Whereas a data warehouse is a place where users can explore data limitlessly, run ad hoc queries, and drill down for details at will, data warehousing is the process of collecting data for the data warehouse. Data warehousing involves the following steps :

— Extract, model and assemble data from the operational systems;

— Transform operational data to a user-focused view;

— Distribute and manage data changes to the warehouse and

— Provide access to data through decision support or executive information system.

A schematic view of data warehousing and data mining is provided in Fig.4.7.

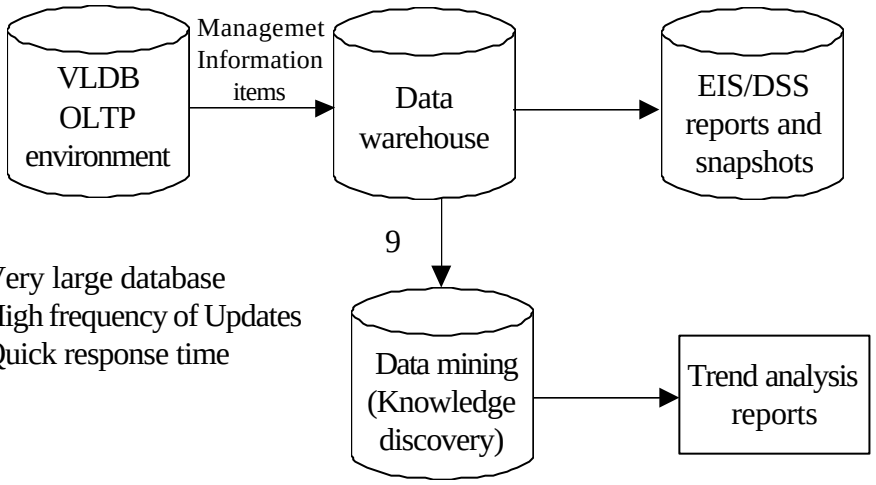

Fig. 4.7 : A schematic of data warehousing and data mining

The discovery of previously unknown information from large databases is known as data mining. Once data warehouses are created, the data stored there is 'mined'. Data mining, which is also called 'knowledge discovery', consists of using automated tools to search for hidden patterns of data. Some tools include statistical methods (e.g., regression, discriminant analysis), pattern recognition (e.g. neural networks) and database segmentation. These tools require substantial computing power and extremely high-speed data retrieval.

11.6 ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Artificial intelligence (AI) is a group of related technologies that attempt to develop machines to emulate human-like qualities such as learning, reasoning, communicating, seeing and hearing.

In the early days of its development a computer was called "electronic brain" and since then efforts have been made to empower it with capabilities comparable with them of the human brain.

However, there is a basic difference in the functioning of the computer and the human brain. Whereas a computer processes numbers, the human brain works on symbols. A conventional computer programme uses algorithms, i.e., a well-defined step-by-step procedure to solve a problem, while the human thought process is not

algorithmic and is based on symbolic processing. In symbolic processing, information is represented using symbols rather than numbers. Human intelligence or thought process relies on heuristic methods for information processing. Heuristics of the human thought process can be represented as shown in Fig. 4.8.

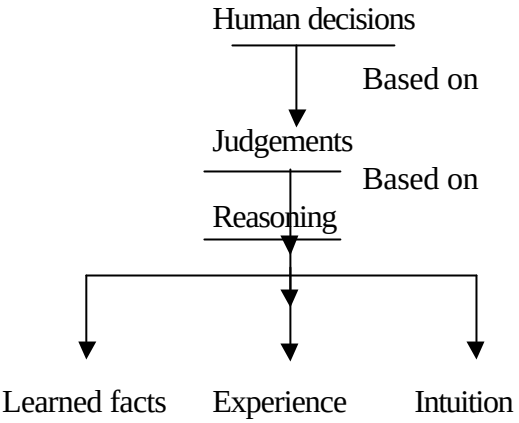

Fig. 4.8 : Heuristics of human thought process

The main areas of AI are :

- Robotics
- Perception systems
- Fuzzy logic
- Neural networks
- Genetic algorithm
- Natural language processing
- Expert systems

Robotics is a field that attempts to develop machines that can perform work normally done by people. The machines themselves are called robots. Perception systems are sensing devices that emulate the human capabilities of sight, hearing, touch and smell. Clearly, perception systems are related to robotics, since robots need to have some sensing capabilities. Fuzzy logic is a method of dealing with imprecise data and uncertainty, with problems that have many answers rather than one. Unlike classical logic, fuzzy logic is more like human reasoning. It deals with probability and credibility. That is, instead of being simply true or false, a proposition is mostly true or mostly false, or can be more true or more false.

Fuzzy logic principles are applied in neural networks. Neural networks use physical electronic devices or software to mimic the neurological structure of the human brain. Genetic algorithms refer to programmes that use Darwinian principles of random mutation to improve themselves.

In *natural-language processing,* a user can interact with a computing system using everyday language rather than a structural command language. Natural language systems have an interface called natural language interface which translates the natural language into the application languages (such as SQL). For a language to be understood there are two aspects, namely, syntax and semantics which play a vital role. Syntax represents the rules for combining words to form phrases, clauses and sentences while semantics refers to word meanings and the way the concepts are expressed. A natural language interface is capable of analyzing the syntax and

semantics of a language. However, there are a number of limitations with the natural language systems at present which have hindered their widespread use. With time the systems are likely to get more refined and popular.

An *expert system* can be defined as AI programmes designed to represent human expertise in a specific domain. Typical expert system application areas include diagnosis, planning, instruction and managment, monitoring and design.

At the heart of an expert system is the *knowledge base* which contains facts (basic data), rules and other information pertaining to a particular domain. Facts represent accepted knowledge in a certain field (normally possessed by experts) and rules – a collection of 1F – THEN statements also called "rules of the thumb" (heuristics) for drawing inferences : 1F such-and-such is true, THEN assume that so-and-so is true.

Unlike databases which store explicit information, knowledge bases with their 1F-THEN rules can infer additional information not directly stored in the basic data. L1SP (list processing) and PROLOG (programming in logic) are the languages most commonly utilized by knowledge bases.

The overall process of developing an expert system is called *knowledge engineering*. In order to map human knowledge to computer knowledge, one must understand the nature of knowledge. Knowledge can be grouped into: procedural knowledge, declarative knowledge, semantic knowledge and episodic knowledge. Procedural knowledge includes the skills an individual knows how to perform. It refers to knowing how to do something. Declarative knowledge represents information that can be verbalized or expressed. It states facts about the world. Semantic knowledge is a deep-level knowledge that reflects the cognitive structure, organization and representation. Episodic knowledge represents information that has been chunked or compiled episodically. It refers to experiential information of an expert.

The expert systems are made up of three basic components: the knowledge base, an inference engine and a user interface. A specialist called knowledge engineer interviews the experts for the domain and encodes their knowledge (in the form of rules) into the knowledge base. The inference engine includes programmes that are used to control how the rules in the knowledge base are used or processed. The user interface facilitates communication or interaction between the expert system and the end user. The components of an expert system have been represented schematically in Fig.4.9.

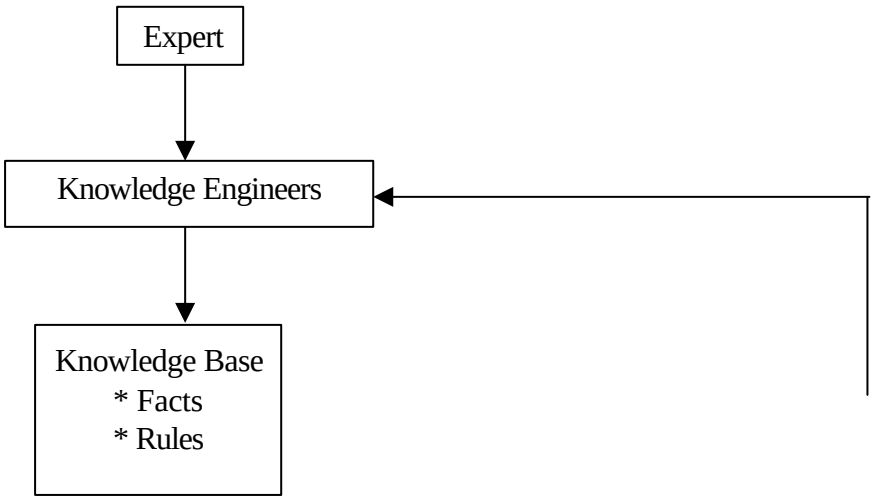

Fig. 4.9 : Components of an expert system

On a request being made by a user, for a specific information, the system looks for data and rules in the knowledge base. The inference engine decides which rules to fire and how they should be applied and also when to complete the querying process and provide the answer. User interface may prompt more input from a user to help the system respond effectively to the query.

The use of expert systems is growing rapidly. However, human experts are likely to remain in control using the expert systems as a job aid.

Self Check Exercises

2) What is the difference between a knowledge base and a database ?

3) What is artificial intelligence (AI) ? List some of the areas of its application.

Note: i) Write your answer in the space given below.

ii) Check your answer with the answers given at the end of this unit.

.....

.....

.....

.....

.....

.....

.....

11.7 SUMMARY

Database techology is making tremendous advances and a variety of database systems have appeared on the scene in recent years. Distributed database systems which provide geographical data independence permit large databases to be distributed over a number of locations transparent to the users. Data warehousing and data mining techniques are augmenting DSS/EIS capabilities. Artificial intelligence has opened up new frontiers in database systems. Knowledge engineering, with expert systems and knowledge bases is providing new tools catalysing socio-economic development. The needs of the emerging information society critically depend on advancements in database technology.

11.8 ANSWERS TO SELF CHECK EXERCISES

1) The advantages of distributed databases are as given below :

- Increased reliability and availability
- Improved data integrity and data administration
- Improved access time
- Modular growth

2) A database stores explicit information while a knowledge base with IF-THEN rules can infer additional information not directly stored in the basic data. In a knowledge base, inference rules allow relationships to be derived from the data.

3) Artificial intelligence refers to computer programmes which emulate human intelligence i.e. programmes enabling computers to perform tasks that require intelligence when performed by humans. Examples of such tasks are visual perception, understanding natural language, game-playing, theorem-proving, medical diagnosis and engineering design.

AI methods have been successfully applied in areas like : knowledge-based systems (expert systems, knowledge bases, etc.), robotics, computer vision, machine translation, neural networks and others.

11.9 KEYWORDS

Artificial Intelligence : A branch of computer science that is attempting to develop systems to emulate human-like qualities such as learning, reasoning, communicating, seeing and hearing.

Data Mining : Also called 'knowledge discovery', it is the Computer-assisted process of sifting through and analysing vast amounts of data (generally in a data warehouse) in order to extract meaning and discover new knowledge.

Data Warehouse : A specialised database that is optimized for management queries. Data is extracted from online transaction processing systems cleaned and customised for searching and analysis.

Expert System : A system with a knowledge base consisting of data and rules that enable users to make decisions as effectively as an expert.

Knowledge Base : A knowledge base is an expert system's database of knowledge about a particular subject. This includes relevant facts, rules and procedures for solving problems. The basic unit of knowledge is expressed as an IF-THEN-ELSE rule.

Transaction : An event that occurs as a part of doing business such as deposit, withdrawal, refund and so on. It refers to a logical unit of operation on a database which may specify a query or addition, deletion or modification of records.

Transparent : In computing, pertains to a process or procedure involving a user without the latter being aware of its existence.

11.10 REFERENCES AND FURTHER READING

Barbara C. McNurlin. Ralph H. Spragu (1989). Information Systems Management in Practice. New Jersey : Prentice-Hall.Inc.

CISMOD 93. Proceedings of the International Conference on Information Systems and Management of Data (1993). Indian National Scientific Documentation Centre (INSDOC): New Delhi

Curtin Dennis P., Kim Foley, Kunal Sen, Cathleen Morin (1999). Information Technology – The Breaking Wave. New Delhi : Tata McGraw-Hill Publishing Company Limited.

Fred. R. MeFadden, Jeffrey A. Hoffer (1988). Database Management. California : The Benjamin/Cummings Publishing Company. Inc.

Jeffrey, D. Ullman (1991). Principles of Database Systems. New Delhi : Galgotia Publications (P) Ltd.

Karn, L. McGraw, Karan Harbison-Briggs (1989). Knowledge Acquisition: Principles and Guidelines. New Jersey : Prentice-Hall. Inc.

Linda. A. Murray, John T.E. Richardon (1989). Intelligent Systems in a Human Context. Oxford : Oxford university Press.

O'Brien, James A (1997). Introduction to Information Systems. Irwin/McGraw Hill.

Post, Gerald V. (2000). Database Management Systems. New Delhi : Tata/McGraw-Hill.

Shepherd, Robert D (2001). Introduction to Computers and Technology. New Delhi : Crest Publishing House.

Su Stanley, Y.W (1988). Database Computers : Principles, Architectures and Techniques. Singapore: McGraw-Hill Book Company.

Williams Brain K, Stacey C. Sawyer, Sarah E. Hutchinson (1999). Using Information Technology. Irwin/McGraw-Hill.