DKY26 - MANAGEMENT DECISION SUPPORT SYSTEM

Unit I Decision support System

Decision concept – Steps – Decision Support System – Components – Characteristics – Clarifications and applications.

Unit II Model Management

Modeling process – types of models – optimization simulation – Heuristic Descriptive – Predictive model case – modeling languages – model directory – model base management system model execution – integration and command processing – model package.

Unit III Data Management System

Data Base – Sources of Data – Data directory – Data Structure and Data Base Language – Query Facility Data Management System – DBMS as DSS development Tool.

Unit IV Dialog Management

User Interface – Graphics – Multimedia – visual interactive modeling – natural language processing – speech recognition and understanding – Issues in user interface.

Unit V Development of decision support system

Development process – software and hardware – data Acquisition – Model acquisition – dialog development – Integration – Testing and validation – Training and Implementation.

References:

1. Ranks – Decision support system
2. Janakiram & Sarukesi – Business Support Systems.
3. Fraim Turban – Decision support and expert System.

UNIT I

Decision Support System

Concept of Decision-Making

Decision-making is a cognitive process that results in the selection of a course of action among several alternative scenarios. Decision-making is a daily activity for any human being. There is no exception about that. When it comes to business organizations, decision-making is a habit and a process as well.

Effective and successful decisions result in profits, while unsuccessful ones cause losses. Therefore, corporate decision-making is the most critical process in any organization. In a decision-making process, we choose one course of action from a few possible alternatives. In the process of decision-making, we may use many tools, techniques, and perceptions. In addition, we may make our own private decisions or may prefer a collective decision.

Usually, decision-making is hard. Majority of corporate decisions involve some level of dissatisfaction or conflict with another party.

Decision-Making Process

Following are the important steps of the decision-making process. Each step may be supported by different tools and techniques.

Step 1: Identification of the Purpose of the Decision

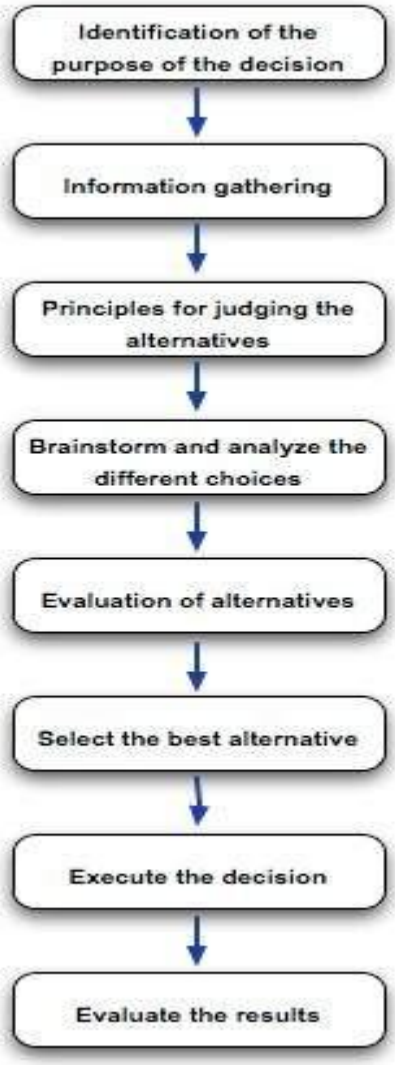
In this step, the problem is thoroughly analyzed. There are a couple of questions one should ask when it comes to identifying the purpose of the decision.

- What exactly is the problem?
- Why the problem should be solved?
- Who are the affected parties of the problem?
- Does the problem have a deadline or a specific time-line?

Step 2: Information Gathering

A problem of an organization will have many stakeholders. In addition, there can be dozens of factors involved and affected by the problem.

In the process of solving the problem, you will have to gather as much as information related to the factors and stakeholders involved in the problem. For the process of information gathering, tools such as 'Check Sheets' can be effectively used.



Step 3: Principles for Judging the Alternatives

In this step, the baseline criteria for judging the alternatives should be set up. When it comes to defining the criteria, organizational goals as well as the corporate culture should be taken into consideration.

As an example, profit is one of the main concerns in every decision making process. Companies usually do not make decisions that reduce profits, unless it is an exceptional case. Likewise, baseline principles should be identified related to the problem in hand.

Step 4: Brainstorm and Analyze the Choices

For this step, brainstorming to list down all the ideas is the best option. Before the idea generation step, it is vital to understand the causes of the problem and prioritization of causes.

For this, you can make use of Cause-and-Effect diagrams and Pareto Chart tool. Cause-and-Effect diagram helps you to identify all possible causes of the problem and Pareto chart helps you to prioritize and identify the causes with the highest effect.

Then, you can move on generating all possible solutions (alternatives) for the problem in hand.

Step 5: Evaluation of Alternatives

Use your judgment principles and decision-making criteria to evaluate each alternative. In this step, experience and effectiveness of the judgment principles come into play. You need to compare each alternative for their positives and negatives.

Step 6: Select the Best Alternative

Once you go through from Step 1 to Step 5, this step is easy. In addition, the selection of the best alternative is an informed decision since you have already followed a methodology to derive and select the best alternative.

Step 7: Execute the decision:

Convert your decision into a plan or a sequence of activities. Execute your plan by yourself or with the help of subordinates.

Step 8: Evaluate the Results:

Evaluate the outcome of your decision. See whether there is anything you should learn and then correct in future decision making. This is one of the best practices that will improve your decision-making skills.

Decision Support System

DSS is an information system application that helps in decision-making. DSS is used in planning and analyzing alternatives. DSS differs from most traditional information system in that each DSS is distinct from the other information system and is specifically made for managers.

All though it is used by managers it is part of organization’s MIS. A DSS is prepared for a specific managerial task and special problem and thus its use is limited to that problem. Decision support systems tend to be designed to serve management control level and strategic planning level managers.

The elements of DSS include a database, a model base and software providing interactive dialogue facility for a manager. The data in the database is a combination of master files, and data from external sources. The second component of DSS is a library of models to manipulate and analyze the data in the desired ways.

The third component is the user interface through this the user can communicate with the DSS. The physical interface generally consists of a terminal attach to the mainframe computer either directly or by telephone. DSS can be differentiated from MIS in terms of its processing capabilities. whereas MIS process data to convert it into information, DSS processes information.

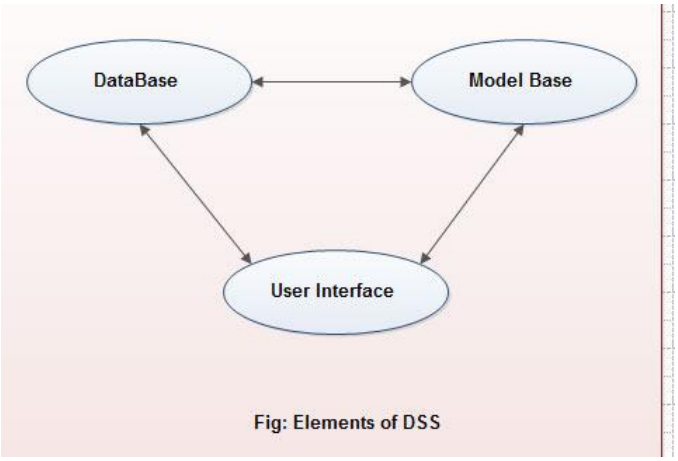


Fig: Elements of DSS

to support the decision making process of a manager. e.g. a salary information system provides information to every employee regarding his basic salary, allowances and deductions. However if any employee wants to make deposits in some schemes for income tax rebates he can make use of DSS. DSS helps the user to decide in which scheme how much he should invest to get maximum benefits.

Decision Support System characteristics and Components

Decision Support System are interactive information systems that depend on a (having different things working together as one unit) set of user-friendly hardware and software tools to produce and present information that is targeted to support the management in the decision-making process. The decision support systems help management decision-making by combining data, fancy (or smart) (related to careful studying or deep thinking) models and user-friendly software into a single powerful system that can support semi-

structured or (without rules, schedules, etc.) decision-making. The decision support system is under user control, from early beginning to final putting into use and daily use. Decision support system helps to close the information gap to enable managers to improve quality of their decisions. While MIS is carefully thought believed useful for structured decisions, DSS is carefully thought about to be more useful for decisions at the strategical/strategic levels, where decision-makers are often angrily stood up to with complex decisions which are beyond their human abilities to (creation/combination)e properly the factors involved. DSS refers to a class of systems, which support in the process of decision-making and does not always give a decision itself. These systems can be used to validate decision by performing sensitivity analysis on different guidelines of the problem. read more about information technology and human resources meaning

While developing decision Support System, the focus must be on identifying a problem and a set of capabilities that users consider useful in arriving at decisions about that problem. While developing DSS, therefore, care must be taken to make sure that the Decision Support Systems possess the following desirable characteristics:

1. Should aid the decision-maker in decision-making.
2. Should be able to address semi/un-structured decision-making situations.
3. Should support decision-makers particularly at tactical/strategic levels.
4. Should be able to create general-purpose models, simulation capabilities and other analytical tools available to decision-maker.
5. Should enable users to use DSS without assistance from MIS/technical professionals.
6. Should be readily adapted to meet information requirement for any decision environment.
7. Should provide mechanism to enable rapid response to a decision-maker’s request for information.
8. Should have the capability to interface with corporate database.
9. Should be flexible to accommodate variety of management styles.
10. Should facilitate communication between/among various levels of decision-making.
11. Should have in-built flexibility and ability to evolve as user-sophistication grows.
12. Using of interactive methods are better advised.

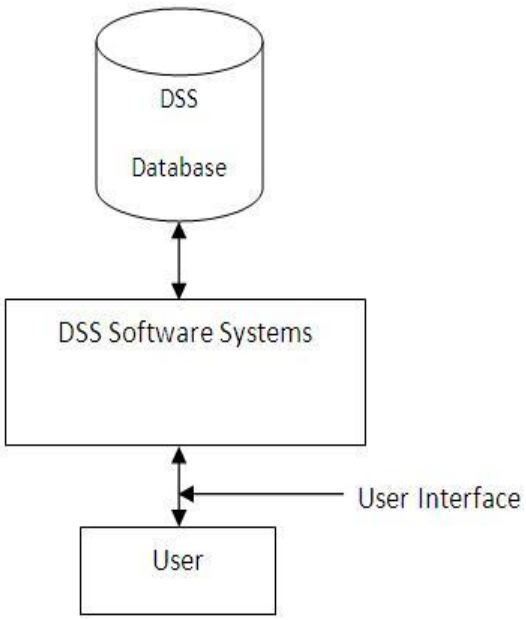
Components of Decision Support Systems (DSS)

Decision support systems consist of three main components, namely database, software system and user interface.

1. DSS Database: It contains data from various sources, including internal data from the organization, the data generated by different applications, and the external data mined form the Internet, etc. The decision support systems database can be a small database or a standalone system or a huge data warehouse supporting the information needs of an organization. To avoid the interference of decision support system with the working of operational systems, the DSS database usually contains a copy of the production database.

2. DSS Software System: It consists of various mathematical and analytical models that are used to analyze the complex data, thereby producing the required information. A model predicts the output in the basis of different inputs or different conditions, or finds out the combination of conditions and input that is required to produce the desired output.

A decision support system may compromise different models where each model performs a specific function. The selection of models that must be included in a decision support system family depends on user requirements and the purposes of DSS. Note that the DSS software contains the predefined models (or routines) using which new models can be built to support specific type of decisions.



Some of the commonly used mathematical and statistical models are as follows:-

- **Statistical Models:** They contain a wide range of statistical functions, such as mean, median, mode, deviations etc. These models are used to establish, relationships between the occurrences of an event and various factors related to that event. It can, for example, relate sale of product to differences in area, income, season, or other factors. In addition to statistical functions, they contain software that can analyze series of data to project future outcomes.
- **Sensitivity Analysis Models:** These are used to provide answers to what-if situations occurring frequently in an organization. During the analysis, the value of one variable is changed repeatedly and resulting changes on other variables are observed. The sale of product, for example, is affected by different factors such as price, expenses on advertisements, number of sales staff, productions etc. Using a sensitivity model, price of the product can be changed (increased or decreased) repeatedly to ascertain the sensitivity of different factors and their effect on sales volume. Excel spreadsheets and Lotus 1-2-3 are often used for making such analysis.
- **Optimization Analysis Models:** They are used to find optimum value for a target variable under given circumstances. They are widely used for making decisions related to optimum utilization of resources in an organization. During optimization analysis, the values for one or more variables are changed repeatedly keeping in mind the specific constraints, until the best values for target variable are found. They can, for example, determine the highest level of production that can be achieved by varying job assignments to workers, keeping in mind that some workers are skilled and their job assignment cannot be changed. Linear programming techniques and Solver tool in Microsoft excel are mostly used for making such analysis.
- **Forecasting Models:** They use various forecasting tools and techniques, including the regression models, time series analysis, and market research methods etc., to make statements about the future or to predict something in advance. They provide information that helps in analyzing the business conditions and making future plans. These systems are widely used for forecasting sales.
- **Backward Analysis Sensitivity Models:** Also known as goal seeking analysis, the technique followed in these models is just opposite to the technique applied in

sensitivity analysis models. In place of changing the value of variable repeatedly to see how it affects other variables, goal seeking analysis sets a target value for a variable and then repeatedly changes other variables until the target value is achieved. To increase the production level by 40 percent using the backward sensitivity analysis, for example, first, the target value for the production level can be set and then the required changes to made in other factors, such as the amount of raw material, machinery and tools, number of production staff, etc., to achieve the target production level.

3. DSS User Interface: It is an interactive graphical interface which makes the interaction easier between the DSS and its users. It displays the results (output) of the analysis in various forms, such as text, table, charts or graphics. The user can select the appropriate option to view the output according to his requirement.

A manager, for example, would like to view comparative sales data in tabular form whereas an architect creating a design plan would be more interested in viewing the result of analysis in a graphical format. The present-day decision support system built using the Web-based interface provides its users some special capabilities like better interactivity, facility for customization and personalization, and more ease of use.

Application

- One example is the clinical decision support system for medical diagnosis. Other examples include a bank loan officer verifying the credit of a loan applicant or an engineering firm that has bids on several projects and wants to know if they can be competitive with their costs.
- DSS is extensively used in business and management. Executive dashboard and other business performance software allow faster decision making, identification of negative trends, and better allocation of business resources.
- A growing area of DSS application, concepts, principles, and techniques is in agricultural production, marketing for sustainable development. For example, the DSSAT4 package, developed through financial support of USAID during the 80's and 90's, has allowed rapid assessment of several agricultural production systems around the world to facilitate decision-making at the farm and policy levels. There are, however, many constraints to the successful adoption on DSS in agriculture.

- DSS are also prevalent in forest management where the long planning time frame demands specific requirements. All aspects of Forest management, from log transportation, harvest scheduling to sustainability and ecosystem protection have been addressed by modern DSSs. A comprehensive list and discussion of all available systems in forest management is being compiled under the COST action Forsys • A specific example concerns the Canadian National Railway system, which tests its equipment on a regular basis using a decision support system. A problem faced by any railroad is worn-out or defective rails, which can result in hundreds of derailments per year. Under a DSS, CN managed to decrease the incidence of derailments at the same time other companies were experiencing an increase.

Benefits

- Improves personal efficiency
- Speed up the process of decision making
- Increases organizational control
- Encourages exploration and discovery on the part of the decision maker
- Speeds up problem solving in an organization
- Facilitates interpersonal communication
- Promotes learning or training
- Generates new evidence in support of a decision
- Creates a competitive advantage over competition
- Reveals new approaches to thinking about the problem space
- Helps automate managerial processes

Advantages of Decision Support Systems (DSS)

- **Time savings.** For all categories of decision support systems, research has demonstrated and substantiated reduced decision cycle time, increased employee productivity and more timely information for decision making. The time savings that have been documented from using computerized decision support are often substantial. Researchers, however, have not always demonstrated that decision quality remained the same or actually improved.
- **Enhance effectiveness.** A second category of advantage that has been widely discussed and examined is improved decision making effectiveness and better decisions. Decision quality and decision making effectiveness are however hard to document and measure.

Most researches have examined soft measures like perceived decision quality rather than objective measures. Advocates of building data warehouses identify the possibility of more and better analysis that can improve decision making.

- **Improve interpersonal communication.** DSS can improve communication and collaboration among decision makers. In appropriate circumstances, communications-driven and group DSS have had this impact. Model-driven DSS provides a means for sharing facts and assumptions. Data-driven DSS make "one version of the truth" about company operations available to managers and hence can encourage fact-based decision making. Improved data accessibility is often a major motivation for building a data-driven DSS. This advantage has not been adequately demonstrated for most types of DSS.
- **Competitive advantage.** Vendors frequently cite this advantage for business intelligence systems, performance management systems, and web-based DSS. Although it is possible to gain a competitive advantage from computerized decision support, this is not a likely outcome. Vendors routinely sell the same product to competitors and even help with the installation. Organizations are most likely to gain this advantage from novel, high risk, enterprise-wide, inward facing decision support systems. Measuring this is and will continue to be difficult.
- **Cost reduction.** Some researches and especially case studies have documented DSS cost saving from labor savings in making decisions and from lower infrastructure or technology costs. This is not always a goal of building DSS.
- **Increase decision maker satisfaction.** The novelty of using computers has and may continue to confound analysis of this outcome. DSS may reduce frustrations of decision makers, create perceptions that better information is being used and/or creates perceptions that the individual is a "better" decision maker. Satisfaction is a complex measure and researchers often measure satisfaction with the DSS rather than satisfaction with using a DSS in decision making. Some studies have compared satisfaction with and without computerized decision aids. Those studies suggest the complexity and "love/hate" tension of using computers for decision support.
- **Promote learning.** Learning can occur as a by-product of initial and ongoing use of a DSS. Two types of learning seem to occur: learning of new concepts and the development of a better factual understanding of the business and decision making environment. Some DSS serve as "de facto" training tools for new employees. This potential advantage has not been adequately examined.

- **Increase organizational control.** Data-driven DSS often make business transaction data available for performance monitoring and ad hoc querying. Such systems can enhance management understanding of business operations and managers perceive that this is useful. What is not always evident is the financial benefit from increasingly detailed data.

Regulations like Sarbanes-Oxley often dictate reporting requirements and hence heavily influence the control information that is made available to managers. On a more ominous note, some DSS provide summary data about decisions made, usage of the systems, and recommendations of the system. Managers need to be very careful about how decision-related information is collected and then used for organizational control purposes. If employees feel threatened or spied upon when using a DSS, the benefits of the DSS can be reduced. More research is needed on these questions.

Disadvantages of Decision Support Systems (DSS)

Decision Support Systems can create advantages for organizations and can have positive benefits, however building and using DSS can create negative outcomes in some situations.

- **Monetary cost.** The decision support system requires investing in information system to collect data from many sources and analyze them to support the decision making. Some analysis for Decision Support System needs the advance of data analysis, statistics, econometrics and information system, so it is the high cost to hire the specialists to set up the system.
- **Overemphasize decision making.** Clearly the focus of those of us interested in computerized decision support is on decisions and decision making. Implementing Decision Support System may reinforce the rational perspective and overemphasize decision processes and decision making. It is important to educate managers about the broader context of decision making and the social, political and emotional factors that impact organizational success. It is especially important to continue examining when and under what circumstances Decision Support System should be built and used. We must continue asking if the decision situation is appropriate for using any type of Decision Support System and if a specific Decision Support System is or remains appropriate to use for making or informing a specific decision.
- **Assumption of relevance.** According to Wino grad and Flores (1986), "Once a computer system has been installed it is difficult to avoid the assumption that the things it can deal with are the most relevant things for the manager's concern." The danger is that once

DSS become common in organizations, that managers will use them inappropriately. There is limited evidence that this occurs. Again training is the only way to avoid this potential problem.

- **Transfer of power.** Building Decision Support Systems, especially knowledge-driven Decision Support System, may be perceived as transferring decision authority to a software program. This is more a concern with decision automation systems than with DSS. We advocate building computerized decision support systems because we want to improve decision making while keeping a human decision maker in the "decision loop". In general, we value the "need for human discretion and innovation" in the decision making process.
- **Unanticipated effects.** Implementing decision support technologies may have unanticipated consequences. It is conceivable and it has been demonstrated that some DSS reduce the skill needed to perform a decision task. Some Decision Support System overload decision makers with information and actually reduce decision making effectiveness.
- **Obscuring responsibility.** The computer does not make a "bad" decision, people do. Unfortunately some people may deflect personal responsibility to a DSS. Managers need to be continually reminded that the computerized decision support system is an intermediary between the people who built the system and the people who use the system. The entire responsibility associated with making a decision using a DSS resides with people who built and use the system.
- **False belief in objectivity.** Managers who use Decision Support Systems may or may not be more objective in their decision making. Computer software can encourage more rational action, but managers can also use decision support technologies to rationalize their actions. It is an overstatement to suggest that people using a DSS are more objective and rational than managers who are not using computerized decision support.
- **Status reduction.** Some managers argue using a Decision Support System will diminish their status and force them to do clerical work. This perceptual problem can be a disadvantage of implementing a DSS. Managers and IS staff who advocate building and using computerized decision support need to deal with any status issues that may arise. This perception may or should be less common now that computer usage is common and accepted in organizations.

- **Information overload.** Too much information is a major problem for people and many DSS increase the information load. Although this can be a problem, Decision Support System can help managers organize and use information. Decision Support System can actually reduce and manage the information load of a user. Decision Support System developers need to try to measure the information load created by the system and Decision Support System users need to monitor their perceptions of how much information they are receiving. The increasing ubiquity of handheld, wireless computing devices may exacerbate this problem and disadvantage.

UNIT II

Model Management

Building Model-Driven Decision Support System (MDSS)

As the name itself explains, a model-driven decision support system utilizes a model to solve problems or aid in decision making. A model can be statistical, financial, mathematical, analytical, simulation or optimization. **A model-driven DSS may employ a single model or a combination of two or more models, depending upon the specific needs of its users**. Simple models provide basic functionality while combination of two or more models lets users analyze complex data.

Model-driven DSS are generally not data intensive. Rather they use parameters entered by decision makers and help them analyze a situation. They generate optimal solutions that are consistent with time and resource constraints. The scope of model-driven DSS is huge and can be further enhanced by integrating web-based applications.

When developing proprietary MDSS, it's important to understand modeling and analytical tools, their working and scope. Building model-driven DSS requires a considerable level of expertise. Managers and DSS analysts need to work closely to develop an efficient system, which is scalable, versatile and easy to integrate and use.

Modeling Decision Situations

Model-driven DSS can be used to aid decision making in a variety of situations. It can assist managers in making:

- Credit and lending decisions
- Product demand forecasting
- Budgeting decisions
- Marketing decisions
- Production forecasting decisions
- Resource allocation decisions
- Project planning
- Investment decisions

Each MDSS has a clear objective and specific purpose. It deploys a model. Consequently, a lot of thought goes into deciding what models should be included in a model-driven DSS. MDSS usually carries out sensitivity analysis or 'what if' analysis.

However, the users must remember that the system doesn't make a decision. It only generates alternatives that are to be analyzed and assessed by decision-makers.

How to Build a Model-Driven DSS ?

The most important aspect of a model-driven DSS is the model it uses for decision making. This means that the selection of a model is the most crucial step in building an MDSS. So, how you go about it? Let's understand:

Modeling

Modeling is the process of identifying an appropriate model for a prospective model-driven decision support system. It goes through following phases in a chronological manner, beginning from problem identification:

Once modeling is done, it's vital to validate the selected model, to ensure it works well and generates appropriate results. Model validation is done by comparing model's output an the actual behavior of the event.

Assumptions & Forecasts

Assumptions are predictions or best guesses. Each model has certain assumptions about the time and risk involved in a particular situation. These results are tested through sensitivity or what if analysis.

Assumptions play an important role in defining a problem and identifying and dealing with uncertainty. Decision makers form a hypothesis and attempt to predict results. Basis the outcome, a hypothesis is either accepted or rejected. Model-driven DSS are designed assuming any of the analyses – static and dynamic.

i. **Static Analysis:** This type of analysis doesn't take into consideration the long term response of a system. It takes a single snapshot of a situation and assumes that it will remain stable all through and won't change. Static analysis is done when a situation in which company makes a decision is static in nature.
ii. **Dynamic Analysis:** Dynamic analysis is testing a program or a software system in real-time. This method considers that the situation changes over time, due to any reason, such as cost, rules and regulations, time, etc.

What kind of analysis needs to be conducted depends upon the situation. Decision makers and DSS analysts must identify whether it is appropriate to assume certainty, uncertainty or risk in a situation.

Certainty	**Uncertainty**	**Risk**
▪ When adequate information about a situation is *available*. ▪ Models based on certainty/static analysis tend to yield optimal solutions.	▪ When information available is vague, unpredictable or unreliable. ▪ It's important to acquire more information to find an appropriate model.	▪ When information is missing. ▪ What if analysis is carried to aid decision making.

Model Types

As mentioned earlier, each model-driven DSS works on some kind of model or a combination of models. Therefore, knowing about various models pays off. A DSS deploys one or combination of below models:

i. **Explanatory/Descriptive Model:** Describes and explains why something is the way it is and why and how it works.

ii. **Contemplative Model:** Forecasts results or outcomes that may be produced from a specific set of parameters.

iii. **Algebraic Model:** A high-level modeling system for solving complex equations. It is employed to optimize a variable or equation. The best part is that it can handle several simultaneous equations.

A DSS with any one of above models performs a single function whichever it is meant to do while a DSS with multiple models is a complete system to perform all three tasks, including:

- Identifying relationships between variables
- Forecasting results based on changes or parameters
- Deciding to what extent a variable can be manipulated

Models

1. **Accounting and Financial Models**

These model-driven decision support systems aid in decision making in various situations related to accounting and financial management. The examples include:

- **Break-Even Analysis:** A DSS with break-even analysis model aids managers in determining a break-even point for a product. It helps establish a what-if selling price and analyzing the relationship between various related components – prices, marketing spend and profits. The process begins by assuming fixed and variable costs. Profit is set at zero. It helps determine a break even cost of a product at which the company is neither loss nor makes profit.
- **Budget Financial Model:** DSS with budgeting model is typically an enterprise-wide application. Many companies use such systems for budget planning and forecasting.
- **Pro Forma Financial Statements:** A DSS with this model summarizes the anticipated financial results for a specific time period in future. Costs are estimated based on past data, gross sales are predicted and profit or loss is then calculated on these relationships.
- **Ratio Analysis:** This helps a business in evaluating its financial statements. Ratio analysis makes financial data more meaningful, by showing logical relationships between data.

2. Decision Analysis Models

The main job of decision analysis models is to identify and evaluate alternatives with their respective pros and cons. The decision makers then evaluate all the alternatives and pick the one that they think is the best. The aim of decision analysis techniques is to:

- Decompose and restructure the problems
- Help decision makers gain in-depth understanding of the problem
- Separate facts and figures from preferences and priorities
- Help users study the performance of decision alternatives
- Avoid citing priorities that don’t help in decision making

The following are various types of decision analysis models:

f. **Analytical Hierarchy Process (AHP):** It's a multi-criteria decision technique that combines quantitative and qualitative factors when evaluating alternatives. The analytical hierarchy process begins with developing a hierarchical representation of a problem, with the overall objective on the top, decision alternatives at the bottom and relevant attributes and selection criteria in between.

After you write decision alternatives at the bottom, you need to compare the alternatives by generating relational data. Consistency ratio is calculated after comparing relative priority of each attribute. The alternatives with the highest priorities and topmost objectives are then displayed.

g. **Decision Trees:** As the name suggests, a decision tree uses a tree-like flowchart of decisions, draw from left to right, with further branches explaining their consequences, cost involved, event outcomes and utility. The aim is to identify the most appropriate strategy to reach a goal. A decision tree has three types of nodes

- Choice node: represented by a square
- Chance node: represented by a circle
- End node: represented by a triangle

The nodes and decision rules are the building blocks of decision trees. The decision trees are simple to understand, offer valuable insights, determine the best and worst scenarios and can be combined easily with other decision techniques.

h. **Multi-Attribute Utility Analysis (MAUA):** Multi-attribute utility analysis gives much importance to attribute weights. The information is provided about each decision choice on each attribute. A decision maker then perceives the utility of usefulness of a decision alternative in terms of its attributes. This method is generally used when the attributes of an alternative are certain.

i. **Influence Diagrams:** It's a diagrammatic representation of a decision situation, to express the precise nature of relationships between variables. It uses geometric shapes to represent various elements.

- A decision variable is represented by a rectangle.
- An intermediate variable is represented by a circle.
- A result or outcome variable is represented by an oval.

Forecasting Models

Forecasting models form an integral part of a large number of decision support systems. Their main job is to predict the value of interrelated variables at some point of time in future. The two main types of forecasts are:

- **Short run forecasts:** where the prediction will be used anytime soon mainly in deterministic models
- **Long run forecasts:** where the prediction is used for long term investment/planning decisions

Forecasting may include ambiguity as factors on which decisions depend are uncontrollable and dynamic in nature. This means that the accuracy of data and time taken in making near-perfect predictions matter a lot.

The following are various types of forecasting models:

l. **Naïve Exploration:** As is explained by the name itself, naïve exploration is not a sophisticated prediction. Rather it is simple forecasting that provides limited accuracy. The technique is implemented using a spreadsheet.

m. **Judgment Methods:** The predictions or forecasting are based on the perceptions and opinions of experts instead on hard data. It's a subjective estimate used for long-run forecasts where external environment plays a critical role. The results are not very accurate.

n. **Moving Average:** Used for short-run forecasts, the predictions are based upon the historical values. DSS with this model is inexpensive and easy to use.

o. **Exponential Smoothing:** Used for short-term forecasts, it alters the historical data mathematically to better reflect the assumptions of a decision maker. Similar to moving average model but claims to obtain better results using exponential smoothing.

p. **Time Series Extrapolation:** This method takes into account the economic variables that are measured at consecutive intervals of time. It is believed that the knowledge of past behavior of the variable at successive intervals of time will help understand the behavior of the variables in future better.

q. **Regression and Econometric Models:** These types of forecasting models make use of linear and multiple regressions to establish cause and effect relationships. These methods are considered more powerful than time-series

but also complex at the same time. They are complex because they use sophisticated models and include more variables. The results obtained are more accurate.

Network and Optimization Models

Network and optimization models are integrated into a DSS when decisions regarding resource allocation, project control, location, scheduling, transportation, distribution, size, shortages, multinational cash flow management, inventory management and distribution and network need to be made. For example:

- The best location for an operation or manufacturing
- The resources needed to carry out the operations
- Most suitable aircraft route to transport products

Network and optimization models typically use linear regression technique, which falls in the class of mathematical programming tool. Using this technique, problem solvers can find the best set of values that minimizes or maximizes a specified calculated formula. A linear programming situation consists of six elements, including:

- Decision variables, the value of which we try to find by applying the model
- Objective function, a mathematical expression showing linear relationship between the goal and decision variables
- Coefficients of objective function, the variables that express the pace at which the value of the objective function alters (increases or decreases) when the values are included in the equation
- Constraints, the linear inequalities reflecting the fact that the resources are limited
- I/O (Input-Output) Coefficients, the coefficients of constraints which indicate the pace at which a given resource is utilized/depleted
- Capacities, which express the minimum resources needed

Remember that it's the managers who determine what 'best' means for them.

Simulation Models

DSS with simulation models conduct experiments to identify conditions or situations that approximate the actual conditions. These models are utilized to solve a number of problems, including

- Manpower planning and assignment

- Inventory control
- Reliability and replacement
- Sequencing and scheduling
- Stock-in and stock-out
- Queuing and congestion

Simulation models:

- Try to imitate reality
- Perform what-if analysis
- Are descriptive tools for forecasting
- Repeat experiments to obtain an optimized estimate of impact of certain actions
- Aid in solving extremely complex problems
- Form elementary relationships and interdependencies among variables
- Are made for one problem and aren't suitable for another problems
- Reduce the time taken in decision making

Simulation Methodology

The process goes through a number of steps, beginning from problem identification and ending at evaluating the results.

Simulation models are of following types:

oo. **Probabilistic:** In this method, experts conceptualize one or more independent variables as a probability distribution of values

pp. Time dependent: Also known as discrete simulation, it takes into account the exact time of the occurrence of an event

qq. **Visual Simulation:** This method uses visuals and animations of results to foster quick and deeper understanding.

Modeling Languages and Spreadsheets

As models are computerized software program, a number of programming languages can be used for coding. Typically the languages used are C++ and Java. Moreover, the decision support systems make use of spreadsheets, allowing users to

- Write values
- Manipulate data

- Apply mathematical and statistical formulas
- Create graphs and visuals
- Prepare, consolidate and sort reports

There are numerous software packages available for model-driven decision support systems. However, you need to carefully select a package. You must ensure that it meets all your specific needs. Reputable packages allow you to create your own models and manipulate the existing ones.

Building a customized model-driven DSS is a complex, time consuming and expensive process. However, the end decision of buying a package or develop a DSS lies with you.

Building Knowledge-Driven Decision Support System and Mining Data

"*Knowledge should be shared. It only grows by sharing*."

This phrase finds its importance in today's highly competitive and economically turbulent business world. Unless knowledge is shared among employees, it doesn't take an organization anywhere. It's important to share and manage it, in order to foster innovative thinking, develop and train employees and evolve into an ever-growing company.

Like it's important to share knowledge within the organization, it's equally important to determine what to share with whom. Not all details can be shared with everyone. This means that it is absolutely necessary to decide knowledge sharing rules and regulations, so that it can be used effectively and appropriately.

So, how do you think, knowledge is shared and distributed within an organization? What it takes to ensure its effective allocation and circulation? How do you automate the access and sharing of information?

Automating Knowledge Sharing

Implementing knowledge-driven decision support system is one of the best ways to capture, process and store and share knowledge among employees. The information can be easily accessed by the user to resolve a variety of problems, issues or concerns.

Before the development of knowledge-driven DSS, employees with high intellect had to perform knowledge-intensive tasks. An expert in a particular area would know how to approach a problem and go about it. Similarly, knowledge-based DSS asks relevant questions, offers suggestions and gives advice to solve a problem. The only difference is that it's automated and speeds up the whole process.

What is a Knowledge-Driven DSS ?

A knowledge-driven DSS

- is a computer-based reasoning system
- that provides information, comprehension and suggestions to users
- to support them in decision-making.

It's an integration of computerized business intelligence tools and technologies customized to the needs and requirements of an organization. So, the focus is on

- Identifying specific knowledge sharing and distribution needs of a company
- Setting objectives that need to be attained with a knowledge-driven DSS
- The selection of appropriate tools and technologies
- Understanding the nature of work and decision-making performed by its potential users
- Selecting data mining techniques

Key Terms and Concepts

A computer-based reasoning system is similar to any other type of decision support system when it comes to their architecture. But it turns into a knowledge-drive decision support system when artificial intelligence technologies, management expert systems, data mining capabilities and other communication mechanisms are integrated.

Before we dig deeper, let's learn about few important terms and concepts used alongside knowledge-drive decision support system. This will help gain an in-depth understanding of such support systems.

1. **Expertise:** A knowledge-drive DSS comes with a specific problem-solving expertise. This expertise is based upon three components:
    - Knowledge in a particular domain and associated symptoms and signs
    - Understanding of the relationships between varied symptoms of a problems
    - Skills, ways or methods of solving the problem
2. **Expert System:** A computer system that imitates the decision making capability of a human expert is called an expert system or an artificial intelligence system. It is designed to solve problems by
    - Using if-then rules
    - Reasoning about knowledge

- Drawing inferences from facts and rules

3. **Knowledge Discovery and Data Mining:** These are interrelated terms used for the process of extracting valuable knowledge and discovering patterns, in order to transform the knowledge into easily comprehendible structure for further use. Data mining is a buzzword but a misnomer. This is because data mining is a process of collection, storing and analysis of data and not finding patterns. Knowledge discovery goes through a series of steps:
    - Selection
    - Pre-processing
    - Transformation
    - Data mining
    - Interpretation
4. **Development Environment:** It's the environment in which a decision support system is developed. It typically includes software for creating a DSS and knowledge base. The development environment may vary in size, depending upon production/development needs.
5. **Domain Expert:** A domain expert is a subject matter expert who has expertise/authority in a particular domain. A domain expert is an integral part of the team working on developing a decision support system.
6. **Knowledge Engineer:** A technical expert who integrates knowledge into a computer system when developing a decision support system, in order to solve complex problems that require human expertise.
7. **Knowledge Acquisition:** It is extraction/mining of knowledge from various sources, such as experts, databases and external programs.
8. **Knowledge Base:** It is the collection and storage of structured (facts, rules, regulations, characteristics, functions, procedures and relationships) and unstructured information that will be used by a DSS in decision making.
9. **Interface Engine:** It is a software system to simplify the conception and development of application interfaces between application systems. Typically, it's a middleware application to transform, route and translate messages between various communication points.
10. **Heuristic:** It's an approach to discovery and problem solving by employing practical methods. These methods may not be optimal but can help achieve immediate goals.

It's important to be familiar with technical jargons that experts in this field use, in order to gain a deeper understanding of knowledge-driven DSS.

Characteristics of Knowledge-Driven Decision Support Systems

A knowledge-driven DSS is different from conventional systems in the way knowledge is extracted, processed and presented. The former attempts to emulate human reasoning while the latter responses to an even in a predefined manner. The main characteristics of knowledge-driven decision support systems are:

- These systems aid managers in solving complex problems.
- These systems allow users to interact with them during the process of decision making.
- The recommendations made by these systems are based on human knowledge.
- These systems use knowledge base that's engineered keeping in mind the nature of problems they will solve.
- These systems aid in performing limited tasks.
- These systems use heuristic technique of problem solving.

Managing Knowledge-Driven Decision Support System Projects

Knowledge-driven decision support systems are expert systems that are developed when decision-making cannot be supported using traditional methods. A knowledge-driven DSS project goes through various stages and can be difficult to manage. It's important to be committed to monitor the development of a knowledge-driven DSS.

Development Stages

- Domain identification (Choosing a subject matter)
- Conceptualization (idea formation, feasibility testing and commencement)
- Formalization (beginning with development officially)
- Implementation (completion and execution)
- Testing (fixing errors and modifications)

It's important to monitor project development throughout very closely. It's a collective effort of knowledge engineers, domain experts, DSS analysts, users and programmers. And a project manager keeps track of the scope, time, quality and budget, to ensure optimum allocation of resources and creation of a quality product. A project manager is a person responsible accomplishing the pre-decided objectives of a project.

Knowledge-Driven DSS Examples

Here are few examples of successful and popular knowledge-driven decision support systems:

- **XCON (eXpert CONfigure):** This expert system was built to decide the components required to build a complete operational system. Its job was to determine the spatial relationships among the components. The DSS configured VAX computers and was known as the largest rule-based knowledge-driven system for years.
- **TAXADVISOR:** As the name suggests, TAXADVISOR assisted attorneys by collecting client data and suggested actions that clients need to take to settle their financial profile. Its job was to aid attorneys in taxation and estate planning for clients with estates greater than $175,000.
- **Life Insurance Selection Expert System:** The expert system helped Meiji Mutual Life Insurance Company, one of the oldest insurance companies in Japan, deploys XpertRule to select the most suitable insurance product for an individual from among myriads of products.

Data Mining and Creating Knowledge

Before, data mining systems came into existence, businesses had statisticians studying data. They would look at the data, formulate a hypothesis and carry out a test to approve or disapprove it. But a data mining software doesn't need to establish a hypothesis to be approved or disapproved. Rather it works in 'discovery mode' and looks for patterns.

Data Mining Models

There are two types of data mining models that can be deployed:

- Predictive Model: This data mining model predicts which prospects are likeliest to respond to a particular stimulus. It forecasts clear results based on patterns identified from known results. It takes into account the people who have already responded to a similar or same stimulus.
- Descriptive Model: This model describes patterns in existing data to create significant demographics subgroups, which can then be used for target marketing.

Data Mining Tools and Techniques

There are a large number of tools and techniques used to extract/mine data. Which technique is to be used depends on the type of data to be extracted.

1. **Case-based Reasoning**

   Case-based reasoning tools are used to determine the distance between or relationship among various components. A problem solved using this tool goes through 5 stages:

   - Presentation – the problem is described and entered into the system
   - Retrieval – the system matches it with the cases stored in the system
   - Adaptation – the system matches the retrieved closest-matching case and the problem to generate a solution
   - Validation - the solution then goes through a validity test and is justified if the user gives a positive feedback
   - Update – the valid solution is accepted and added to the case base in the system

2. **Fuzzy Query and Analysis**

   Fuzzy query and analysis is a data mining tool follows the mathematical concept for 'fuzzy logics – the logic of uncertainty' to determine results that are close to a particular criterion. Users can then pick one, depending upon his or her understanding.

3. **Data Visualization**

   As the same suggests, this helps analysts visualize complex relationships in multi-dimensional data. The benefit is that this tool graphically represents relationships among components from different perspectives. Statistical tools, such as regression, classification or cluster analysis are a part of this tool.

4. **Genetic Algorithms**

   Similar to linear programming models, genetic algorithms conduct random experiments by selecting the genes (variables whose values are to be identified) and their values at random to find the fitness function. The software will also combines and mutates genes to find optimized value.

Data Mining or Knowledge Extraction Process

Knowledge extraction is the process of identifying relationships between various components or symptoms. It's about making the best use of data. Data mining or knowledge creation proceeds through a number of stages:

- Setting objectives
- Selecting data to be mined

- Run feature or cluster analysis to qualify data
- Selecting and applying an appropriate data mining tool
- Discover and apply knowledge to solve a problem

Data Mining Examples

Now you know what data mining is and how knowledge is extracted from data collection and analysis, let's take a look at few data mining tools that companies are using.

- Siemens, a German multinational conglomerate and the largest engineering company in Europe, uses a decision support system that uses case-based reasoning tools, helping their technical support service staff answer the questions from current enquiry. The DSS uses results of previous enquiries as cases and retrieves close matching cases when a problem is entered into the system.
- ShopKo, a Wisconsin based chain of retail stores selling clothing, footwear, bedding, jewelry, beauty, house ware, etc uses a data mining project to find that the sale of film does not result in the sale of the camera. Rather it's the sale of camera that affects the sale of films.
- Firstar Bank, now U.S. Bank, headquartered in Minneapolis, US, used data mining tools to determine the customers who were interested in knowing about their new products. They started doing the target emails, which increased the response rate.
- American Century Investments is a independent and privately controlled investment management firm, headquartered in Kansas City, United States. The firm makes use of data mining techniques to know who all from among their customers will be interested in buying their other products. They use the results to cross-sell their products.

Evaluating Development Packages

Whenever you decide to develop or buy a knowledge-driven decision support system software application, it's important to consider following criteria:

- **Development Features:** Input rules, customizability, capabilities and maintenance
- **Scalability:** Ease of integration with other existing hardware and software, web technologies, operating systems
- **Ease of Use and Installation:** The ease with which end user will be able to work on it

- **Security:** Safety of data and company information
- **Cost:** Cost of technology, cost of development, maintenance cost

Knowledge-driven decision support systems help businesses solve problems and make decisions. However, a caution should be used when employing it. It doesn't outsmart human intellect; rather it aids decision making.

Building Web-Based and Inter-Organizational Decision Support Systems

Problem solving and knowledge management go hand-in-hand. Together they have become one of the most important aspects of organizational decision making. Managers around the world realize that much of their organizations' value depends on their ability to gather, analyze and manage knowledge and use it to solve problems.

To accelerate the process of transforming information into knowledge and applying it to resolve problems, organizations use resources in addition to human intellect. These resources help them discover patterns, identify context where these patterns work and analyze alternative solutions.

A computerized decision support system does all the work and aids managers in making the right choice. However, with the arrival of the internet and modern communication technologies, the organizational decision support systems deploy web-based technologies. The new science of gathering and distributing information adds efficiency to DSS, ultimately helping managers make more appropriate decisions.

The best part is that the web-based decision support systems can be knowledge-driven, communications-driven, model-driven, document-driven or data-driven or hybrid. Web technologies can be integrated with any type of decision support system. Since internet technologies are known for effective, faster and safer information distribution, the web-based DSS can be used across the organization and by two or more organizations where there is a need for information sharing.

Key Terms

Before we move on and get into the intricacies involved in building a web-based decision support system, it's important to understand these terms:

- **W3 or www:** It stands for the World Wide Web, which in itself is an information system of interlinked hypertext documents that are accessed via the internet. Commonly, hypertext documents are called web pages. Generally W3 and internet are

used interchangeably. But they are not same. Internet is a system of interconnected computer networks whereas W3 is one of the services for these networks. The latter is a collection of images, text documents and other resources that are linked by URLs and hyperlinks.

- **Web-based DSS:** Web-based DSS is a computerized system to deliver information for decision support using a 'thin-client' browser like Internet Explorer or Netscape Navigator. The computer server that hosts DSS application is linked to the user's computer by a network using TCP/IP (Transmission Control Protocol/Internet Protocol).
- **Inter-Organizational DSS:** An inter-organizational DSS is a system that's used by organizations working together on a particular problem or project. A DSS used by a company's investors, bankers, stockholders, suppliers and customers is an example of an inter-organizational DSS. Web technologies drive communication and information distribution.
- **Intranet:** It is a private network that can be accessed only by an organization's staff. A DSS used across the organization facilitates information sharing through intranet.
- **Extranet:** It's not a private network but allows controlled access to information to vendors, partners, suppliers and customers.
- **TCP (Transmission Control Protocol):** It's the core protocol of the IPS (Internet Protocol Suite), to provide trustworthy, controlled, prearranged delivery of digital information units between applications running on hosts, over an IP network. TCP is a protocol on which emailing, W3, FT (file transfer) and remote administration applications rely on.

Designing and Developing Web-Based DSS

Most companies simply integrate a web browser on their existing decision support systems, in the name of upgrading it and deploying a so-called web-based DSS. The results, most likely, are unsatisfactory and ambiguous. This is because the web technologies are integrated without carrying a feasibility analysis. When a DSS is initially designed, it focuses on the current requirements of its users. Ideally, managers shouldn't expect it to be efficient in other areas. So, the question remains - what goes into designing and developing a web-based decision support system?

Here is the step-by-step process explaining how a web-based system is designed and developed. However, this is a standard process and any of the below mentioned steps can be

removed during system development, depending on the understanding and preferences of developers, analysts and users.

1. **Problem Identification:** This is universal because without having to know what you want a web-based DSS to do for you, it's impossible to move ahead – let alone getting things right. Brainstorm with your team members to identify the problems you want to resolve using a DSS.
2. **Conceptualization:** Next step is to conceptualize the idea on which your DSS will be based. Ask yourself these questions:
    - What is it going to do?
    - How is it doing to do?
    - What techniques will it use?
    - Who's going to use it?
    - What processes do you want it to carry?
3. **Feasibility Analysis:** Once you have a clear idea of a prospective decision support system, you must assess it. Try to uncover
    - Its strengths and weaknesses
    - Opportunities and threats
    - Resources required to build it
    - Prospects for success

    In addition to this, a feasibility analysis sheds light on a project's operational, economical, technological, scheduling and legal viability.
4. **System Development:** Once the prospective system has passed your feasibility test, you can begin with system development. Remember that system development doesn't just require technical expertise; rather it's a collaborative effort of managers, DSS analysts, developers and finance professionals. The focus must be on:
5. **UI Development:** Developing a user interface remains the most crucial aspect of any computer-based system. And it becomes more important when a prospective system will be used by individuals of various levels of sophistication and technical know-how. Control, memory aids and suggestions based on user's history are important components of a user interface, when it comes to a web-based system.
6. **DSS Architecture:** User interface is just a part of DSS architecture and thus, is not separated from it. However, we mentioned it separately because UI is like a

middleman that fosters communication between the system and the user. Arguably, it's the most important aspect of DSS architecture.

Typically a web-based DSS is built on three or four tier architecture. One, user sends a request through a web browser using HTTP to a web server. Two, the web server then uses a program or a script to process the request. Three, the script may link to a model, processes a database request or format a document. Four, the web browser where the user sent a request displays results most suited to his/her query.

7. **Tool Selection:** HTML (Hyper Text Markup Language) is the most commonly used tool for developing web-based DSS. HTML is not a programming language, but can be used for input and output from a decision support which is programmed in Java or JavaScript. HTML has been basically designed to stipulate the rational organization of documents with hypertext extensions.

   XML (Extensible Markup Language), CGI (Common Gateway Interface Scripts), JavaScript code in HTML pages, Java applets and ActiveX are other tools that can be used to develop web-based DSS. In order to weigh your options, you must consult your developer and determine development costs.

8. **System Implementation:** Once the system is developed, the next phase is to implement it. A number of tools are available that specialize in implementation of DSS. These are ColdFusion and dbProbe.

Potential Problems

When an organization gets on with web-based DSS, there can arise many potential problems, which can be easily resolved. Let's take a look at potential problems and their solutions.

- As the DSS is web-based, it may encounter peak load problems when many managers use the system simultaneously. High performance hardware can resolve this problem.
- Web is stateless and doesn't keep track of configuration settings automatically. At the same time users don't want to reenter the username and password each time they use the system. You must consult your developers or solution providers on how to handle user authorization and authentication.
- Users must learn to use the system rapidly. This is because web technologies keep advancing and there will be a need to upgrade the system on regular basis. Training can help users keep up with the changing technology.

Managing Web-Based and Inter-Organizational Decision Support System

Managing web-based and inter-organizational decision support systems can be daunting. This is because these systems are used by a many individuals for making shared decisions. Despite observing extreme caution when developing a system, there are some real-world issues that are bound to arise. Reengineering business process can be really challenging. It's not an easy task to redesign or reengineer a business process when it involves a huge number of users. Managers in interdependent organizations must consider following issues to ensure effective implementation of a web-based DSS:

- The first major issue is – who will use the system: managers, suppliers, customers or all? When you answer this question, you'll identify the associated questions: whether there will be a need to redefine processes; how are you going to train the users to work on a DSS; what are the chances of its success, etc. If the answers to these questions are ambiguous, your DSS is destined to fail.
- The second major issue is availability of technical talent to develop a web-based DSS. Experts need to determine hardware, software and manpower required to build a system. So, managers need to ask themselves if they have in-house capabilities to work on developing the system. If no, where are they going to find their technology partners?
- Third major issue is legality of the information distributed through DSS to external parties. Consider, if you are likely to face any copyright or privacy issues?

Examples of Web-Based DSS

There are several famous web-based decision support systems, including but not limited to:

- **Microsoft Carpoint:** Demonstrating both data and model-driven DSS, Microsoft Carpoint is a web-based system that allows users to make pair-wise comparisons of car models.
- **TCB Works:** A communication-driven web-based DSS, TCB Works enables people to interact, discuss and make both structured and multi-criteria decisions. Users are prompted to enter username and password, in order to start with the project screen.
- **Fidelity 'Retirement Planning Calculator':** It is a model-driven DSS to help a person decide how much he or she needs to invest each month for retirement.

- **Netscape Decision Guides:** These are model and knowledge-driven web-based decision support systems, offering 25 different decision guides on diverse topics, including choosing pets, bikes and business schools.
- **Stockfinder:** It's a data-driven DSS to help investors identify stocks based on various criteria, including industry type, price and earnings.

Advantages and Disadvantages of Web-based DSS

Web-based decision support systems though are the latest in the class. But they also have a fair share of pros and cons. Let's take a look at their advantages and disadvantages:

Advantages

- Web-based DSS reduce decision making costs to a great extent.
- They have reduced geographical and technological barriers.
- They make it easy for companies to involve their vendors, suppliers and customers in decision making, surveys, etc.
- They have significantly improved the speed of information distribution.
- They provide an excellent way to create and manage a knowledge repository.
- They have reduced end-user training costs as web technologies nowadays are used by almost everyone.

Disadvantages

- Web-based DSS are extremely efficient, due to which users sometimes set unrealistic expectations.
- They may succumb to peak demands and experience load problems.
- Web-based DSS require additional security, which may be expensive.
- Web technologies advance at a very fast pace. It may be difficult to upgrade the system so frequently.
- The knowledge repository may accumulate obsolete reports, if users don't delete old information.

The internet and World Wide Web have created major opportunities to create, manage and share both quantitative and qualitative information while keeping the costs low. Though these technologies don't resolve all the problems but they have contributed significantly to knowledge management and decision making.

Web-based DSS can be very effective if managers are aware of what they want to create and how they want to use it. These systems are expected to evolve and become smarter with the advancement in internet technologies.

UNIT III

Data Management System

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

Characteristics

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics −

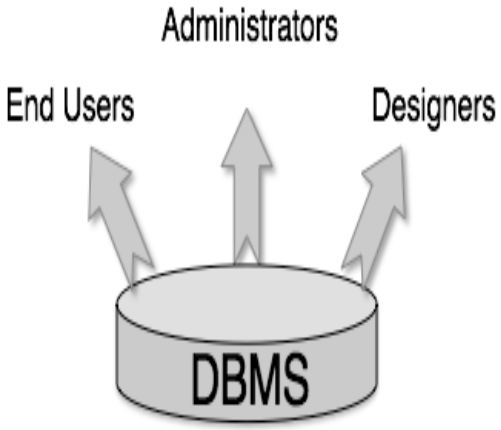
- **Real-world entity** − A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- **Relation-based tables** − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.
- **Isolation of data and application** − A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.
- **Less redundancy** − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.
- **Consistency** − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- **Query Language** − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different

filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.

- **ACID Properties** − DBMS follows the concepts of **A**tomicity, **C**onsistency, **I**solation, and **D**urability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.
- **Multiuser and Concurrent Access** − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.
- **Multiple views** − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.
- **Security** − Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

Users

A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows −



- **Administrators** − Administrators maintain the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.
- **Designers** − Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.
- **End Users** − End users are those who actually reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

Data Dictionary

A **data dictionary** (also called the **metadata**) is the data about the data. It is the self describing nature of the database that provides program-data independence. It is also called as the System Catalog. It holds the following information about each data element in the databases, it normally includes:

+ Name
+ Type
+ Range of values
+ Source

+ Access authorization

+ Indicates which application programs use the data so that, when a change in a data structure is contemplated, a list of the affected programs can be generated.

Data dictionary is used to actually control the database operation, data integrity and accuracy. Metadata is used by developers to develop the programs, queries, controls and procedures to manage and manipulate the data. Metadata is available to database administrators (DBAs), designers and authorized user as on-line system documentation. This improves the control of database administrators (DBAs) over the information system and the user's understanding and use of the system.

Active and Passive Data Dictionaries

Data dictionary may be either active or passive. An active data dictionary (also called integrated data dictionary) is managed automatically by the database management software. Consistent with the current structure and definition of the database. Most of the relational database management systems contain active data dictionaries that can be derived from their system catalog.

The passive data dictionary (also called non-integrated data dictionary) is the one used only for documentation purposes. Data about fields, files, people and so on, in the data processing environment are. Entered into the dictionary and cross-referenced. Passive dictionary is simply a self-contained application. It is managed by the users of the system and is modified whenever the structure of the database is changed. Since this modification must be performed manually by the user, it is possible that the data dictionary will not be current with the current structure of the database. However, the passive data dictionaries may be maintained as a separate database. Thus, it allows developers to remain independent from using a particular relational database management system. It may be extended to contain information about organizational data that is not computerized.

Importance of Data Dictionary

Data dictionary is essential in DBMS because of the following reasons:

- Data dictionary provides the name of a data element, its description and data structure in which it may be found.
- Data dictionary provides great assistance in producing a report of where a data element is used in all programs that mention it.

- It is also possible to search for a data name, given keywords that describe the name. For example, one might want to determine the name of a variable that stands for net pay. Entering keywords would produce a list of possible identifiers and their definitions. Using keywords one can search the dictionary to locate the proper identifier to use in a program.

These days, commercial data dictionary packages are available to facilitate entry, editing and to use the data elements.

Introduction to Data Structures

Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. For example, we have data player's name "Virat" and age 26. Here "Virat" is of **String** data type and 26 is of **integer** data type.

We can organize this data as a record like **Player** record. Now we can collect and store player's records in a file or database as a data structure. For example: "Dhoni" 30, "Gambhir" 31, "Sehwag" 33

In simple language, Data Structures are structures programmed to store ordered data, so that various operations can be performed on it easily. It represents the knowledge of data to be organized in memory. It should be designed and implemented in such a way that it reduces the complexity and increases the efficiency.

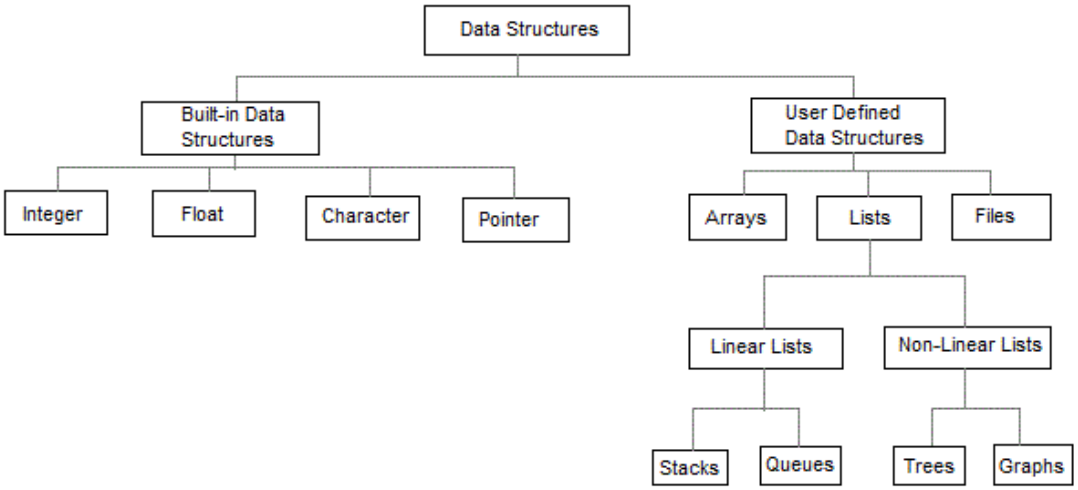
Basic types of Data Structures

As we have discussed above, anything that can store data can be called as a data structure, hence Integer, Float, Boolean, Char etc, all are data structures. They are known as **Primitive Data Structures**.

Then we also have some complex Data Structures, which are used to store large and connected data. Some example of **Abstract Data Structure** are :

- Linked List
- Tree
- Graph
- Stack, Queue etc.

All these data structures allow us to perform different operations on data. We select these data structures based on which type of operation is required. We will look into these data structures in more details in our later lessons.



INTRODUCTION TO DATA STRUCTURES

The data structures can also be classified on the basis of the following characteristics:

Characteristic	Description
Linear	In Linear data structures, the data items are arranged in a linear sequence. Example: **Array**
Non-Linear	In Non-Linear data structures, the data items are not in sequence. Example: **Tree**, **Graph**
Homogeneous	In homogeneous data structures, all the elements are of same type. Example: **Array**
Non-Homogeneous	In Non-Homogeneous data structure, the elements may or may not be of the same type. Example: **Structures**
Static	Static data structures are those whose sizes and structures associated memory locations are fixed, at compile time. Example: **Array**
Dynamic	Dynamic structures are those which expand or shrink depending upon the program need and its execution. Also, their associated memory locations changes. Example: **Linked List created using pointers**

Database Languages

A DBMS must provide appropriate languages and interfaces for each category of users to express database queries and updates. Database Languages are used to create and maintain database on computer. There are large numbers of database languages like Oracle, MySQL, MS Access, dBase, FoxPro etc. SQL statements commonly used in Oracle and MS Access can be categorized as data definition language (DDL), data control language (DCL) and data manipulation language (DML).

Data Definition Language (DDL)

It is a language that allows the users to define data and their relationship to other types of data. It is mainly used to create files, databases, data dictionary and tables within databases.

It is also used to specify the structure of each table, set of associated values with each attribute, integrity constraints, security and authorization information for each table and physical storage structure of each table on disk.

The following table gives an overview about usage of DDL statements in SQL

S.No	Need and Usage	The SQL DDL statement
1	Create schema objects	CREATE
2	Alter schema objects	ALTER
3	Delete schema objects	DROP
4	Reneme schema objects	RENAME

Data Manipulation Language (DML)

It is a language that provides a set of operations to support the basic data manipulation operations on the data held in the databases. It allows users to insert, update, delete and retrieve data from the database. The part of DML that involves data retrieval is called a query language.

The following table gives an overview about the usage of DML statements in SQL:

S. No	Need and Usage	The SQL DML statement
1	Remove rows from tables or views	DELETE
2	Add new rows of data into table or view	INSERT
3	Retrieve data from one or more tables	SELECT
4	change column values In existing rows of a table or view	UPDATE

Data Control Language (DCL)

DCL statements control access to data and the database using statements such as GRANT and REVOKE. A privilege can either be granted to a User with the help of GRANT

statement. The privileges assigned can be SELECT, ALTER, DELETE, EXECUTE, INSERT, INDEX etc. In addition to granting of privileges, you can also revoke (taken back) it by using REVOKE command.

The following table gives an overview about the usage of DCL statements in SQL:

S. No.	Need And Usage	Age
1	Grant and take away priviliges and roles	Grant Revoke
2	Add a comment to the data dictionary	Comment

In practice, the data definition and data manipulation languages are not two separate languages. Instead they simply form parts of a single database language such as Structured Query Language (SQL). SQL represents combination of DDL and DML, as well as statements for constraints specification and schema evaluation.

RDBMS (Relational Database Management System)

The MIS is sustained by database in its endeavour to support the management in decision making. Organization needs that MIS would give them a 'competitive strength'. The need is to handle an on-line operation, mission, control applications and exercise the operational and management control. The need demands a tool to successfully manage both the transaction processing and the decision processing requirements. It also requires the capability of dealing with large number of users who are using, and updating a huge database. The need also demands the use of multiple databases residing on the hardware platforms situated at different location-nearby sites and remote site. The decision-making is essential more in a real time environment where the decision-making process, right from the problem definition to solution, needs to be handled quickly. The business environment is distributed and decentralized requiring real time resources (hardware, software, data, information) sharing with a multifaceted data flow. All this processes require the RDBMS which can serve both the decision support and the transaction processing requirements. With the biggest computer hardware and software capabilities, the mechanism of Relational Database Management System becomes popular (Waman Jawadekar, 2009). Major goal of a relational database design is to generate a set of relation schema that allows us to store information without unnecessary redundancy and also to retrieve information easily.

RDBMS is the database system in which the relationships among different tables are maintained. It corresponds to data as two-dimensional tables called relations or files. Each table contains data on entity and attributes. RDBMS solution is necessary when huge amounts of data are to be stored as well as maintained. A relational data model comprises of indexes, keys, foreign keys, tables and their relationships with other tables. Relational DBMS enforces the rules even though foreign keys support by both RDBMS and DBMS. The most popular RDBMS are MS SQL Server, DB2, Oracle and MySQL.

The RDBMS has five main components:

1. The relational algebra model defines schemas, relations, and declarative specifications of query operations.
2. The SQL compiler includes the parser, definitions for SQL abstract syntax, a denotation specification for SQL in terms of the model, and semantics-preserving SQL optimizations.
3. The SQL execution engine interprets the optimized SQL expression as a series of operations over imperative finite maps. Correctness is established using Hoare-style reasoning relating imperative finite maps to the relations they represent.
4. The B+ tree implementation provides finite map operations for insertion and lookup of key-value pairs, and iteration (amongst others).
5. The storage interface is responsible for deserializing or serialization relations to disk and establishing integrity constraints. The storage manager includes a proof that deserializing the serialized form of a relation R results in R, under the assumption that disk operations do not fail.

Features of RDBMS

1. RDMS supports relational data structure.
2. RDBMS has Data Manipulation Language at least as powerful as the relational algebra.
3. In RDBMS, data is stored in a set of tables.
4. Tables are joined by relational links.
5. RDBMS Reduces Duplication of data in database (Normalization).
6. RDBMS facilitates greater flexibility and efficiency.
7. In RDBMS, each table must have unique references for each record called Primary key.

8. Replicating these into other tables creates the Foreign Key.
9. These foreign keys form the Relationship that links the tables together.
10. Each RDBMS table consists of database table rows. Each database table row consists of one or more database table fields.
11. In RDBMS data needed only be updated once as it would only have been entered once.
12. RDBMS reduces the problems in using Flat file databases.

UNIT IV

Dialog Management

Designing a Decision Support System User Interface

The effectiveness of a computerized system or a software platform depends on its user interface design. Be it a routine software program or a high-end decision support system, user's interaction with the system should be as simple, effective and hassle-free as possible.

It's not a mystery that great decision support systems have brilliant user interface designs. They are engineered to provide superior user experience. Designing a user interface involves attaining complete understanding of user needs and the way they make decisions.

So, the questions arise - what goes into their planning and how to design a good user face? We will come to this. However, before we get into the details of why and how parts, let's first try to understand what user interface actually means.

User Interface

If we look at the literal meaning of the word, interface means the 'crossing point' or 'border'. And in computer science, interface means that part of a computerized system that allows its users to interact with it. It is what users see and use. User interface is important yet it facilitates interaction between the user and the software system without drawing much attention to itself. It is that hidden aspect of a computer program that supports its usability.

User interface may include:

- Commands
- Set of menus
- Graphics
- Icons
- Guidelines
- Acoustics
- Tactile
- Hardware
- Any other presentation

Features of Good User Interface

The most important features of user interface are that it:

- Influences how users interact with the decision support system

- Balances the technicality (functionality of the system) and mentality (user's state of mind)
- Provides users a picture-oriented (visual) way to interact with the system
- Creates flawless communication and interaction between the user and the decision support system
- Reduces errors, increases speed, supports 'good decision making'
- Logical and intuitive at the same time

The quality of a decision support system very much depends on its user interface. This is because it is what a user sees, feels, senses and uses to interact with the software system. Therefore, right from the **screen layout to color, style, symmetry, lines, density, icons, pictures, symbols, commands, input and output display play a decisive role**.

To get things right, DSS analysts and end users need to work together, and cite and resolve potential issues that may arise while using a decision support system.

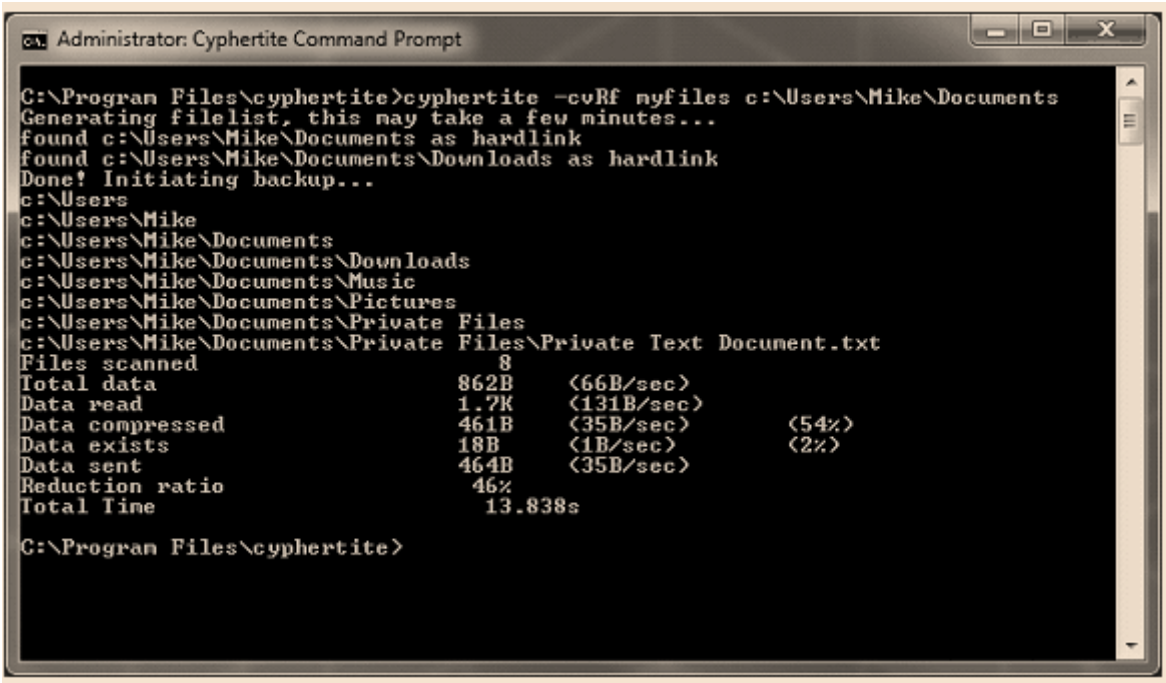
User Interface Styles

As we know, user interface is the space where human-machine interactions take place. This decides how user enters the information and how system presents the outcome. There are different ways to interact with computerized decision making systems.

However, which user interface style is to be used needs to be decided before the development process begins. There are no hard and fast rules about which the best user interface style is. A Decision Support System may use one interface style or a combination of two or more styles.

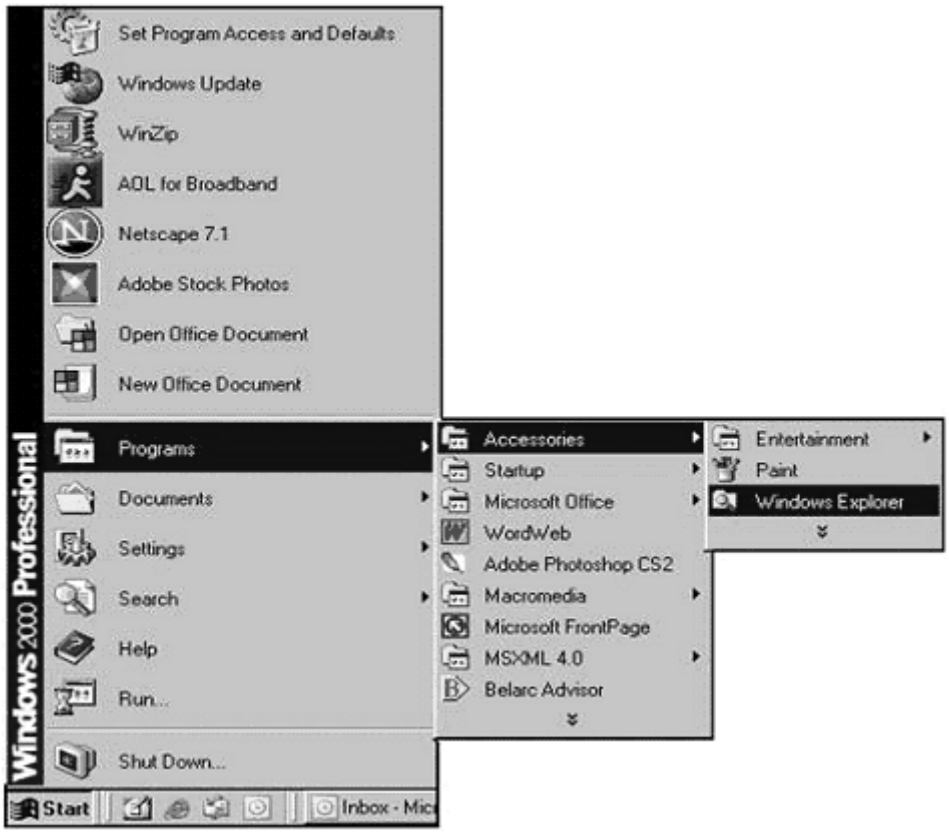
A DSS designer can provide several control sequences to manage or run a software program, depending upon user needs. There are six dominant user interface styles that can be used:

1. **Command-Line Interface:** As the name suggests, the command-line interface made dominant use of commands to set user interaction with the software system. A user would enter the command, such as "run" and the system would execute it. The operating systems used were MS-DOS, UNIX and LINUX. This interface would require a user to enter the command, in order to tell the machine what to do. Though such interfaces were powerful, the user would actually have to learn the commands to make a system work for them, which was definitely restricting.



2. **Menu Interface:** The menu interface offers users a list of functions in the form of choices. The drop-down menu makes it easier for them to select a function that needs to be performed. The best part is the users don't have to learn the commands to make use of software system.

   However, menus are appropriate for simple systems. As complexity increases, the users require several sets of menus to choose from. Even the items in the menu can have sub menus. A good example of such operating system is Windows. But when it comes to building DSS using menu interface, it takes huge time and resources to design and develop the interface and the software itself in the long run.



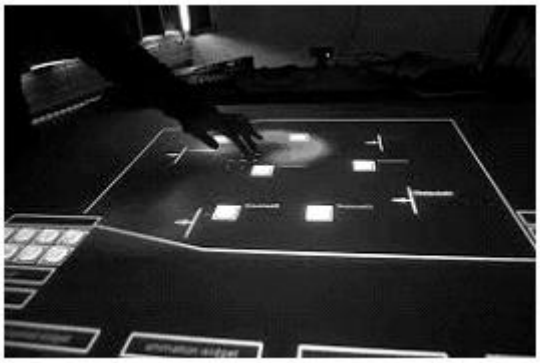
3. **Graphical Interface:** It is an interface system that allows users to give commands through visible objects. You either point or touch the images, icons or symbols to perform an action. The graphic user interface focuses more on multimedia rather than text.

4. **Question-and-Answer Interface:** Such kind of interface allows machine to ask questions and user to enter answers to the questions. It turns into a dialogue when user

continues to answer questions asked by the system. This type of interface is an effort to induce human-to-human interaction through a system. However, the major challenge occurs when the answers fed by the user are unstructured because a computer doesn't understand unstructured responses.

5. **Voice User Interface:** As the name suggests, it makes human-machine interaction possible through speech. A human voice is required to control the machine or to make it perform an action. Voice user interfaces have now become commonplace. These are eyes-free and hands-free interfaces that perform action by recognizing the speech.

6. **Touch User Interface:** This is the most popular and most recent type of user interface. It relies upon the sense of touch and directs the system to perform the chosen action when a user touches a particular visual. In digital environment, it is used alongside voice and graphics user interfaces.



Designing User Interface

An effective user interface is essential for any type of information system. Its design largely determines whether the software system will be used or not. Typically, the focus is always on the expertise of the analysts and designers, which is not the right approach. Although the technical expertise is crucial, the specific needs of the end user need to be understood, accessed and evaluated. With this, the totality of experience working on the system and using it needs to be counted. The right user interface design approach is the first step in developing an efficient decision support system.

ROMC Design Approach - Representation - Operation - Memory Aid - Control Aid

ROMC is an organized approach to design specialized decision support systems, more precisely their user interfaces. It's a systematic move towards designing the representation, operation, memory and control mechanisms of a large decision support system.

Suggested by Ralph Sprague and Eric Carlson in 1982, it focuses on the analysis of the above mentioned four important elements of a user interface. The approach helps identify the essential competencies of a computerized system. Not only is this approach suitable for developing user interface designs but this can also be used for creating screen designs.

Representation

Representation is about presenting information or results for that matter, in a structured way. All decision making activities in an organization take place in a certain environment or context. The representation, in tandem with this context, provides a tangible conceptualization to communicate information to the decision maker or user of the system about the situation.

The representation provides a base to the decision makers backed by concrete information, helping them interpret DSS outputs. The representation can be in the form of a

table, graph, map, chart or a text document and each value on a map or a table communicates decision making context.

Operation

This stage in user interface design focuses on specific tasks performed by/with a decision support system. The operation can involve one activity or many, depending upon the specific needs of the decision maker. A DSS may be used to process data, track market trends, carry analytics or suggest alternatives or perform all the functions.

Memory Aid

How will a DSS operate? On what basis will it produce outputs? What is it going to represent? It must have an access to data to synthesize and analyze. A data warehouse is its memory aid and so for decision makers. So, it must give users a link to data warehouse aid their memory. In addition, it may provide them with links and command shortcuts or sequences to help users control a decision support system.

Control Aid

Control aid is provided to users, so that they can effectively use representations, operations and memory aids. It allows them to edit, redo, undo, save, view, delete any information or representation or the task performed by/with a DSS. They make it easier for the user to interact with and control a decision support system. The commands, however, depends upon the type of user interface.

A user interface design is made up of all these elements and a lot of brainstorming and planning go into it. Designing a user design is not a technical process; rather it is a collective effort of technical and decision making experts. They need to work closely, in order to design a great user interface. This increases the likelihood that a system will be used post development and implementation.

Factors Influencing User Interface Design Success

There are a lot of factors that influence the success of a user interface design. A DSS designer is expected to recognize and consider these factors when designing a user interface for a decision support system. This is done to:

- Eliminate/reduce the fatigue of working on a system
- Reduce the learning time of DSS users
- Reduce the chances of errors made by end users
- Keep users motivated to use the decision support system
- Offer users the ease to recall

So, **here are the factors influencing UI design success**.

- **Execution Time:** Why does a decision maker use computerized system to aid decision making? Obviously, to reduce execution time! As a DSS designer, you must try reduce the execution time for a command given and action performed. Maximize the pace of execution to minimize the wastage of time.
- **Versatility:** A decision support system must be resourceful enough to perform the entire range of tasks that a decision maker needs to perform when making a decision using DSS. Moreover, it should be flexible enough to integrate new tasks whenever needs arises.
- **Adaptability:** A decision support system should be smart enough to adapt according to the most prominent habits of its user. This means it needs to be self-tailoring or customizing in itself. It may seem impractical, but in reality it is not. Rather this is what is expected from a smart decision making system.
- **Learning Time:** A DSS user interface should be simple enough to reduce the learning time of its users, so that they can use it to its full capacity as soon as possible.
- **Uniformity of Command:** As said earlier, a DSS user interface must have a uniform theme throughout. It should offer the same look and feel and command throughout.
- **Quality of Help:** When a decision maker is user a DSS built by you, he or she expects complete on and off line support from you. The success of a DSS depends upon the quality of support offered. Recognize what user may do on/with the DSS and offer self-help manuals both online and offline.
- **Memory Load:** A person has limitations when it comes to remembering numbers. The idea is not to bombard the user with too many statistical or numerical data

interpretations at one time. A good UI design takes the memory load off the user mind.

- **Ease of Recall:** If a user comes back to DSS after long, it must help him/her recall what was done previously. It helps them achieve the same pace in a shortened time.
- **Fatigue:** Mental fatigue occurs because of the complexity of the design. Keep things simple and keep the commands visual so that the user doesn't need to remember anything.
- **Errors:** Anticipate errors that a user may perform when using a decision support system. Provide them the control to reverse the action and help to guide them what to do next.

Designing decision support system user interface is the toughest part of the development cycle. It's the most important element as it establishes the communication between the machine and the human. The use of visual elements and simple screen designs can add a great deal to the success of a DSS.

UNIT V

Development of Decision Support System

Designing and Developing Decision Support Systems

The business intelligence tools or decision support systems aid decision making in an organization. An effective DSS provides you with unbiased data analysis, real time monitoring and rich reporting, supporting you make an informed decision in the least possible time span. A meticulously designed DSS makes use of analytical models, various statistical and econometric tools and of course, human intelligence and insights to support decision making.

This proves the importance of technology, methodology and approach behind designing and development of a decision support system. As a decision maker, you need to be extremely careful in selecting the DSS technology and developmental methodology. You need to get the basics right, in order to get a reliable tool to support diverse information and decision-making needs.

Because **designing and development of a decision support system is a complex process, you may face a number of issues, such as:**

1. **Mismatch between the perspectives of the programmer and decision maker**

   Decision makers, on one hand, have very little knowledge about what it takes to design and develop a decision support system. On the other hand, a programmer or developer may not have a fair idea of how complex business decision making has become in today's highly turbulent times. Technology selection may seem puzzling and ambiguous to decision maker while the preferences, needs and expectations of a decision maker may not fit the process of system design and development. This mismatch between perspectives leads to complexities. But at the same time, this increases the likelihood of creation of an effective and apt DSS.

2. **Identification of specific requirements**

   Most of the times, even decision makers have no clue about what to expect from a decision support system. This is especially the case when managers need a tool to aid in resolving ill-structured or unstructured problems. It is easy to decide what functionality a DSS must possess when problems are routine and structured. Managers can seek help from their team members and conduct brainstorming sessions, in order to chart out the functions and benefits of a decision support system.

3. **Technology selection**

   In today's highly competitive business environment, you need an interactive and responsive decision support system that maximizes your performance. In such a scenario, the selection of technology plays a crucial role. However, it's not always necessary to use the latest version of a particular platform. It completely depends on the objectives you wish to achieve. However, a DSS development company may push you for the latest technology, which will certainly add to your expenses. The rule of thumb is to ask for price quotes from different companies.

4. **Approach to software design and development**

   As said earlier, there is no standard approach or methodology to design the best decision support system. There are three main approaches to system design and development: 1) the traditional system development life cycle (SDLC) approach; 2) rapid prototyping and 3) end-user approach. You may want to choose a methodology that ensures that the final product accomplishes your objectives. But each approach has its pros and cons, making it difficult to choose the best one.

Most of these issues can be resolved by:

- Identifying clear agendas
- Brainstorming with team members
- Communicating the expectations clearly to the programmers
- Educating yourself about various technologies used in DSS development
- Trying to reach at a common platform with the developers

Decision-Oriented Diagnosis

Increasing decision-making effectiveness through changes in how decisions are made should be the major objective for any DSS project. Stabell proposes a decision-oriented design approach for DSS. He argues the pre-design description and diagnosis of decision-making is the key to securing a decision-oriented approach to DSS development.

The diagnosis of current decision-making and the specification of changes in decision processes are the activities that provide the key input to the design of the DSS. Diagnosis is the identification of problems or opportunities for improvement in current decision behavior. Diagnosis involves determining how decisions are currently made, specifying how decisions should be made, and understanding why decisions are not made as they should be. A specification of changes in decision processes involves choosing what specific improvements

in decision behavior are to be achieved. These statements of improvements provide the objectives for the DSS development.

Diagnosis of a decision process involves completing the following three activities:

- Collecting data on current decision-making using techniques such as interviews, observations, questionnaires and historical records;
- Establishing a coherent description of the current decision process;
- Specifying a norm for how decisions should be made.

These activities are interdependent and provide feedback for the analyst. In many DSS development projects it is not feasible to perform a full-scale diagnosis of decision-making.

A shortened study is often necessary due to cost considerations, limited access to managers,or other organizational constraints. As a consequence, DSS analysts should develop the ability to produce diagnosis after only limited exposure to a decision situation.

DSS Audit Plan

	DSS Audit Plan
Step1	Define the decisions, decision processes and related business processes that will be audited. Define the authority of the auditor, purpose of the audit, scope of the audit, timing of the audit, and resources required to perform the audit. Identify a primary contact.
Step 2	Examine the formal design of the process. Diagram the process and specify criteria, etc. Is the design effective and efficient?
Step 3	Examine the actual use of the decision process. Observe the process. Interview decision makers and collect data. Is the process implemented and used as intended?
Step 4	Assess performance of the actual decision process. What works? Can cycle time be reduced? Are decisions appropriate? Timely? Cost effective? Is the process producing value in meeting business objectives? If not, why?
Step 5	Reporting and recommendations. Summarize steps 1-4 in a written report. Discuss what is working well and what needs to be improved. Develop recommendations for improving the process. Hold an exit meeting with decision makers.

Table.1 A DSS Audit Plan.

A related diagnostic activity is conducting a DSS Audit. In general, it can be very useful to audit operational and managerial decision processes. An audit can be a first step in identifying opportunities to redesign business processes and include new Decision Aids and Decision Support Systems in business processes. In some situations, an audit can suggest changes in decision technologies that can improve performance and reduce costs. When an audit is complete the central questions should be how can we do better and what changes should have the highest priority. Table.1 identifies the 5 steps in a DSS Audit.

Diagnosis for some projects focuses on identifying what is assumed by decision-makers in the decision situation and on what is defined by decision-makers as the range of available remedial actions. Focusing on assumptions and actions is appropriate if building a Model- Driven DSS is a possibility, but not when the focus is on a Data-Driven DSS.

Rockart (1979) identified an approach for defining decision-making data needs that is appropriate for Data-Driven DSS and especially Executive Information Systems. Rockart's Critical Success Factors (CSF) Design Method focuses on individual managers and on each manager's current hard and soft information needs. A CSF analysis can be beneficial in identifying "the limited number of areas in which results, if they are satisfactory, will insure successful competitive performance for the organization". If organizational goals were to be attained, then these key areas of activity - usually three to six factors - would need careful and consistent attention from management.

Good diagnosis is difficult, but DSS diagnosis involves skills that can be developed and sharpened. Both managers and MIS staff need to work on completing the diagnosis task. Does diagnosis always provide sufficient information for specifying a DSS? In most cases the diagnosis does provide sufficient information for specifying several alternative designs.

DSS design usually involves a number of difficult tradeoffs. The first tradeoff is whether the DSS should support both the existing process and a prescribed new process. There is also a trade-off in the extent of the capabilities of the DSS and the scope of the process the DSS is designed to support. In most cases the initial version of a DSS focuses on either extensive capabilities for a narrow scope process or few capabilities for a broad scope process.

Prepare a Feasibility Study

Diagnosis of decision-making should be followed by additional initiation and diagnostic activities and preparation of a feasibility study of the technical and economic

prospects related to developing a DSS. This study should occur prior to actually committing resources to developing a proposed DSS. What should be included in a DSS Feasibility Study? This is a common question. An outline for an extensive feasibility study report is included at the end of this chapter. The outline has 15 sections on topics like DSS Scope and Target Users, Anticipated DSS Impacts, Benefits, Risks and Mitigating Factors.
Shorter, less comprehensive studies and reports are usually prepared for small scope DSS projects.

At this point a decision is made between purchasing an application package and in-house development. Packaged DSS applications are often quite versatile and are usually less expensive to implement than in-house development. Packaged solutions are also often faster to implement.

Choose a Development Approach

As noted in the overview, three approaches to DSS development are discussed in the literature and used by practitioners. The approaches or methodologies have been called a variety of names. Essentially we begin by focusing on decisions and decision processes in the decision-oriented design steps, then a project manager or an end-user implement a more or less structured development methodology.

Figure 4.1 shows a recommended process hierarchy for DSS design and development. The process begins with decision-oriented diagnosis and feasibility analysis and then moves to in-house or outsourced development of the proposed Decision Support System using one of three development approaches. We will examine these alternative approaches.

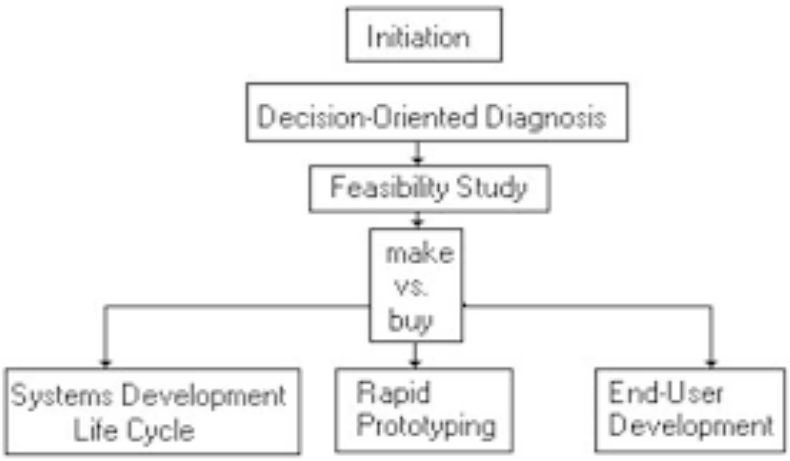


Figure 4.1. A DSS Systems Design and Development hierarchy.

Systems Development Life Cycle Approach

The systems development life cycle (SDLC) approach is based on a series of formal steps, including the following seven steps:

1) Confirm user requirements;
2) Systems analysis;
3) System design;
4) Programming;
5) Testing;
6) Implementation; and
7) Use and Evaluation.

Although different versions of SDLC vary in the precise number of steps and in the detailed definitions of those steps the above steps illustrate the approach. Decision- oriented design begins to address user requirements, but in SDLC user requirements need to be defined in great detail.

This formal SDLC approach is sometimes called the Waterfall model because of the sequential flow from one step to another. Each formal step concludes with preparation of a written progress report that must be reviewed and approved. Reviewers include both prospective users of the system and developers. For example, in Step 5, prospective users verify that the documented functions and capabilities and the user interface meet their needs. Developers verify that the system's internal interfaces are consistently defined and meet all technical requirements.

When the SDLC approach was first formalized in the mid-1970s, it provided structure and discipline to system developers. It was soon adopted widely for developing large-scale transaction-processing systems. SDLC is especially common when a formal contractual relationship exists between the developers of an application system and its eventual users because it provides written evidence that can be used to arbitrate any disputes.

The development of large, shared Enterprise-Wide Decision Support System is often an undertaking of great complexity. Organizational decision processes are complex and computerizing these systems so many people can share them increases that complexity.

Using a methodology like SDLC provides one way in which business organizations can systematically approach the development of an Enterprise-Wide DSS. When the systems development life cycle approach is used, then project plans must be carefully prepared. When

developing requirements, it is best to start by determining the needs of all potential users, then analysts should identify the outputs that would fulfill those needs. Technical requirements should follow logical requirements, and constraints must be identified for all of the DSS system components. These requirements must be documented carefully and reviewed by the targeted users.

Several alternatives may exist for meeting the needs identified during the requirements and design steps. Each of these should be carefully reviewed and the best one chosen. Another choice to be made concerns the make or buy decision. If in-house development is not chosen, a request-for-proposal [RFP] may be required. During the design stage, technical processes must be managed, people and procedures prepared, and an installation plan developed.

In many situations a full-scale SDLC approach is too rigid for building Decision Support Systems, especially those DSS whose requirements are changing rapidly. User requirements, agreed upon at the first stage of the process, are rigidly specified with SDLC.

Any significant change restarts the entire development cycle, as subsequent requirements documents are based on the agreed upon user needs. Changes are therefore often expensive; in fact, SDLC limits change rather than accommodating it.

Rapid Prototyping

All of the different versions of rapid prototyping accommodate and even encourage changes in the requirements of a proposed Decision Support System. A typical prototyping methodology usually includes five steps:

1. Identify user requirements.
2. Develop a first iteration DSS prototype.
3. Evolve and modify the next iteration DSS prototype.
4. Test DSS and return to step 3 if needed.
5. Full-scale implementation.

Prototyping evolved in response to perceived deficiencies and limitations of the SDLC approach. In a prototyping development approach, DSS analysts sit down with potential users and develop requirements. These requirements are specified in general terms and should evolve from the decision-oriented diagnosis and design. The analyst then develops a prototype of a system that appears to work. DSS analysts use tools such as

Database Management Systems and DSS application generators that support rapid development.

Analysts focus on capabilities rather than resolving problems. A prototype may not resolve how to access a real database, or what "help" screens are needed, and other capabilities that require extensive development time. The prototype is something that users can try out, react to, comment on, and eventually approve with a high confidence level that it meets their needs. Missing features are added later, once users are satisfied with the way the prototype works. Rapid Application Development (RAD) specifies incremental development with constant feedback from potential users. The objective of RAD is to keep projects focused on delivering value and to keep clear and open lines of communication. In most situations, oral and written communication is not adequate for specification of computer systems.

RAD overcomes the limitations of language by minimizing the time between concept and actual prototype implementation.

Once approved, a prototype can be expanded in the development environment or the prototype can be used as a specification for a DSS developed in a language like Java, C or C++. When a prototype is reprogrammed, the prototype serves as a detailed specification that is turned into an operational system. The best prototype development approach is to have the actual prototype evolve directly into the finished product. In this approach the prototype is attached to a database and features are added to it, but it remains written in the high-level tools originally used for prototype development.

Compared with the SDLC approach, prototyping seems to improve user-developer communication. It introduces deliberate flexibility and responsiveness into the development process. Change is no longer something to be avoided; it is built into the process and encouraged. The system that is developed is more likely to meet user needs than is a system developed through SDLC.

Prototyping can extend the development schedule if it is improperly used. Managers and developers are often tempted to "tinker" with a DSS and make changes that do not really improve the usability of the finished product. If managers and developers want to build a useful system and meet project deadlines, then they must manage and control systems development efforts.

End-User DSS Development

End-user development of DSS puts the responsibility for building and maintaining a DSS on the manager who builds it. Powerful end-user software is available to managers and many managers have the ability and feel the need to develop their own desktop DSS.

Managers frequently use spreadsheets, like Microsoft Excel and Lotus 1-2-3, as DSS development tools. Using a spreadsheet package, managers can analyze an issue like the impact of different budget options. Following the analysis, managers select the alternative that best meets their department's needs. Also, managers can develop tools to help them conduct market analyses and make projections and forecasts at their desktop.

The major advantage of encouraging end-user DSS development is that the person who wants computer support will be involved in creating it. The manager/builder controls the situation and the solution that is developed. End-user DSS development can also sometimes result in faster development and cost savings.

End-user DSS development of complex DSS is much less desirable. Managers are paid to manage, not to develop Decision Support Systems. At some point DSS specialists can do the work much better and much faster. Also, managers are not trained to test systems, create documentation, provide for back-up and data security and design sophisticated user interfaces. DSS analysts should help managers develop more complex end-user Decision

Support projects. DSS analysts can help the manager build, document and test the application. Managers need to emphasize the content of the DSS and not become overly involved with extensive DSS development projects.

End-user DSS development is a controversial topic. Information systems staffs have many concerns including:

1. End-users may select an inappropriate software product as a development environment.
2. The end-user may have limited expertise in the use of the product and the IT group may have limited resources to support end-user development.
3. Errors during end-user DSS development are frequent. Even experienced developers can make errors and end-users are likely to overlook the need for checking formulas and auditing the DSS they have developed.
4. Unnecessary databases are sometimes developed by end-users for their DSS. Redundant databases can contain out-dated and inaccurate data.

5. A major quality issue involves testing and limited documentation. End-users often perform only limited testing of DSS they develop; and they have limited experience-documenting applications.
6. End-user databases may be poorly constructed and difficult to maintain.
7. End-users rarely follow a systematic development process.

If an organization's MIS group gets actively involved in supporting end-user DSS development, many of the above problems can be minimized, reduced or eliminated.
Packages used for end-user development can be standardized; end-users can be trained in the use of selected packages; support staff can act as consultants and reviewers; a central database can be maintained for use with end-user applications; and documentation can be encouraged by MIS staff.

An Information Center can provide support for end-users and the Director of the Information Center may be able to manage end-user computing. Services that an Information Center might provide include: software training, user support including answering specific development questions, installation assistance and advice about new systems, and standard setting. SDLC and prototyping approaches require designation of a project manager. So let’s now examine DSS project management issues.

Author
S. Mohamed Imran
Assistant Professor,
Sadakathullah Appa College,
Tirunelveli – 627011