CHAPTER 2
LITERATURE REVIEW

The objective of the literature review is to analyze the various models in order to develop new models, considering the limitations of the existing models. This chapter is concerned with the literature review on the theories and models used in the design of human-computer interaction (HCI), single-objective facilities layout models based on the qualitative criteria, multi-objective facilities layout models dealing with the conflicting and congruent objectives of qualitative and quantitative criteria, and multi-factor facilities layout models handling a number of qualitative and quantitative factors in the objective function. This chapter also contains the literature concerned with the various layout procedures and also the various qualitative and quantitative factors that affect the layout of human-computer interface. Finally, the chapter ends with the brief summary of the chapter.

2.1 THEORIES AND MODELS USED IN THE DESIGN OF HUMAN-COMPUTER INTERACTION

Human-Computer Interaction (HCI) needs both mental and physical activities [Norman, 1986]. Physical activities describe the motor actions performed by the user, such as using the keyboard or mouse, to provide inputs. Mental activity refers to the cognitive processes, such as planning for the execution of system inputs and to interpret and evaluate system outputs.
The guidelines used in the design of HCI may be classified as follows [Shneiderman, 2000].

1. High level theories and models, which offer a framework or language to discuss issues that are application independent.
2. Middle level principles are useful in creating and comparing design alternatives, and
3. Specific and practical guidelines provide helpful reminders of rules uncovered by designers.

The creation of a guidelines document engages the designed community in a lively discussion of input or outputs formats and command sequences, terminology, and hardware devices [Brown, 1988; Galitz, 1994]. Inspirations for design guidelines can also be taken from graphic designs [Tufte, 1983, 1990, 1997; Mullet and Sano, 1995].

The separation between basic principles and more informal guidelines is not a sharp line [Shneiderman, 2000]. However, thoughtful designers can distinguish between psychological principles [Wickens, 1993; Bridger, 1995] and practical guidelines that are gained from experience with specific application. The principles are in harmony with the goal of simplifying the user's-task eliminating human actions when no judgment is required. Users can then avoid the annoyance of
handling routine, tedious, and error-prone tasks, and can concentrate

critical decisions, planning, and coping with unexpected situations
[Sanders and McCormick, 1993]. The basic principles involved in
Interacting with systems are as follows [Shneiderman, 2000].

1. Recognize the diversity. When human diversity is multiplied by
the wide range of situations, tasks, and frequencies of use, the
set of design possibilities becomes enormous. The designer can
respond by choosing from a spectrum of interaction styles.

2. Use of the following eight golden rules of interface design.

   i) Strive for consistency
   ii) Enable frequent users to use shortcuts
   iii) Offer informative feedback
   iv) Design dialogue to yield closure
   v) Offer error prevention and simple error handling
   vi) Permit easy reversal of actions
   vii) Support internal locus of control
   viii) Reduce short-term memory load

3. Prevent errors.

Any theory that could help designers to predict performance for
even limited range of users, tasks, or designs would be a contribution
[Card, 1989]. A theory, taxonomy, or model is involved with the
prediction of subjective satisfaction or emotional reactions.
Researchers from information science, business and management, education, sociology, anthropology, and other disciplines are benefiting and contributing by their study in human computer interaction [National Research Council, 1983; MarchionI and Sibert, 1991].

There are different theories, taxonomies, and models developed by the various authors for the design of human - computer interaction (HCI), viz. the four level approach model, GOMS (Goals, Operators, Methods and Selection rules) and the key - stroke level model, stages of action model, the action identification theory and widget level theory.

2.1.1 THE FOUR LEVEL APPROACH MODEL

Foley and Van Dam developed the four - level approach model in the late 1970's [Foley et al., 1990]. It deals with the four levels of action, such as conceptual level, semantic level, syntactic level, and lexical level. The conceptual level is the user's mental model of the interactive system. The semantic level describes the meanings conveyed by the user's command input and by the computer's output display. The syntactic level defines how the units (words) that convey semantics are assembled into a complete sentence that instructs the computer to perform a certain task. The lexical level deals with device dependencies and with the precise mechanisms by which a user specifies the syntax. This approach is convenient for designers because its top -
down nature is easy to explain, matches the software architecture, and allows for useful modularity during design.

2.1.2 GOMS AND THE KEYSTROKE – LEVEL MODEL

Card, Moran, and Newell, (1980, 1983) proposed the cognitive GOMS (Goals, Operators, Methods and Selection rules) model and keystroke level model for the design of HCI, in particular the design of user interface. Olson and Olson (1990) reviewed the state of the art of cognitive GOMS modeling, discussing several extensions to the basic framework in the research stage of development and pointing the way to several more plausible and useful extensions that could be explored. GOMS model postulates that users formulate goals (edit document) and sub goals (input word), each to which is achieved by using methods and procedures (move cursor to desired location by following a sequence of arrow keys). The elementary perceptual, motor, or cognitive acts, whose execution is necessary to change any aspect of the user's mental state or to affect the task environment, are the operators (press up – arrow keys, move hand to mouse, recall file name, verify that cursor is at end of file). The selection rules are the control structures for choosing among the several methods available for accomplishing a goal (delete by repeated back space versus delete by placing markers at beginning and end of region and pressing delete button). Keiras and Polson (1985) used production rules built on GOMS approach called Soar to describe
the conditions and actions in an interactive text editor. Keiras (1988) outlined several significant gaps in cognitive GOMS model and offered a refinement with natural GOMS Language (NGOML). Soar was used to model learning in the highly interactive task of video game playing [Bauer and John, 1995]. John and Keiras (1996a, 1996b) provided ten case studies of practical applications to compare four GOMS related techniques, viz., CMN - GOMS, KLM (Keystroke - Level - Model) GOMS, NGOMSL (Natural GOMS Language), CPM (Critical Path Model) GOMS.

The Keystroke - level model predicts performance times for error - free expert performance of tasks by summing up the time for key stroking, pointing, homing, drawing, thinking, and waiting for the system to respond. These models design the HCI system for error - free performance expert users, and less emphasis is placed on learning, problem solving, error handling, subjective satisfaction, and retention.

2.1.3 STAGES OF ACTION MODEL

Norman (1986) described the stages of action in the form of descriptive model, when the users go through in using the system. The seven stages of action of HCI system model are as follows [Norman, 1988].

1. Forming the goal
2. Forming the intention
3. Specifying the action
4. Executing the action
5. Perceiving the system state
6. Interpreting the system state
7. Evaluating the outcome

The user forms a conceptual intention, reformulates it into the semantics of several commands, constructs the required syntax, and eventually produces the lexical token by the action of moving the mouse to select a point on the screen. This model leads to identification of the gulf of execution (the mismatch between the user's intentions and the allowable actions) and the gulf of evaluation (the mismatch between the system's representation and the user's expectations). This model needs to execute the action plan and the amount of cognitive effort required can be attributed to the semantic distance (which relates to deciding what to do) and the articulator distance (which relates to a decision about how to do it). A syntactic-semantic model of human behavior was originated to describe programming [Shneiderman, 1980] and was applied to data-base manipulation facilities [Shneiderman, 1981], as well as to direct-manipulation [Shneiderman, 1983]. The direct-manipulation interfaces based on theory of automaticity are expected to reduce these distances and efforts needed as compared to command and menu based interfaces [Hutchins et al., 1985]. This model helps to explore the user interface [Poison and Lewis, 1990].
This model suggests four principles, such as visibility of state and action alternatives, good conceptual model, good mapping that reveals the relationships between stages in the interface, and continuous feedback for good design of HCI.

This model helps to describe user exploration of an interface [Polson and Lewis, 1990]. As users try to accomplish their goals, user failures can occur because of the following four critical points [Shneiderman, 2000].

1. Users can form an Inadequate goal.
2. Users might not find the correct interface components (objects) because of an incomprehensive label or icon.
3. Users may not know how to specify or execute a desired action, and
4. Users may receive inappropriate or misleading feedback.

The later three failures may be prevented by improved design or overcome by time-consuming experience with the interface [Franzke, 1995].

2.1.4 THE ACTION IDENTIFICATION THEORY

Valliacher and Wegner (1987) suggest action identification theory, which deals with highly familiar tasks at goal level (what is to be done) and less-familiar tasks at the action level (how to do). These levels are analogous to the stages of human-computer interaction (HCI) in
Norman’s model. Action identification theory explores the conditions for different sequence of actions from that specified by Norman (1986). Once the goal and actions are identified, then the physical steps are carried out. A good interface minimizes the effort associated with deciding what to do and how to do it. The theory of automaticity describes the psychological mechanism that helps to minimize the effort required at each stage.

The theory of automaticity is involved with fast and effortless automatic processing as well as a step-by-step and effortful time consuming processing [Kahneman and Treisman, 1984]. Automaticity can be acquired by the mechanism called strengthening [Logon, 1988b], which serves to make a connection between a stimulus and response progressively stronger with consistent practice or repetition [Laberge and Samuels, 1974; Shiffrin and Schneider, 1977]. Direct-manipulation interface is designed to simulate routine actions that users are familiar with from their everyday experiences, since it requires to drag one icon (component) to another. Menu based interfaces have relatively weaker linkages, since it requires to click on one icon (component) and then another to perform automatic processing.

Mackenzie et al. (1991) have observed that the motor time required to drag objects is actually greater than that of to select menu items by clicking. Direct manipulation interfaces induce automatic
processing and has been observed to provide performance and learning advantages over alternative interface forms such as menus [Benbassat and Todd, 1993]. The results suggest that performance advantages for direct – manipulation must be attributable to shorter non-motor time (total time minus motor time) in the direct – manipulation interface. One possible reason for such a difference is due to the relative influence of automaticity for different interfaces [Lim et al., 1996]. That is, interface characteristics may influence the achievement of automaticity. Automaticity may also be affected with the other factors such as familiarity with the task and the instruction type.

2.1.5 WIDGET – LEVEL THEORIES

The user interface components such as menus, buttons, scroll bars, and text input fields are known as widgets. A measure of layout appropriateness of widgets can be obtained to guide the designer in a possible redesign. The layout appropriateness refers to the frequently used pairs of widgets should be adjacent, and the left – to – right sequence should be in harmony with the task sequence description.

Models requiring numerous subjective judgments concerned with perceptual and cognitive actions, and the issues concerned with motor actions are expected to yield good results in the layout design of user interface components. Estimates of the perceptual and cognitive
complexities plus the motor actions would be generated [Sears, 1992] for the task under consideration.

The layout appropriateness of the widgets relates to the optimum layout of the facilities in a manufacturing plant. In the problem of widgets layout for the interface design, the facility can be a text or a graphical user interface component (icon).

Users of text retrieval systems often experience difficulties when trying to phrase their information need in the retrieval language of the system [Morten Hertzum and Erlk Frokjaer, 1996]. One reason for this is that it is inherently difficult to give a precise and differentiating description of something one lacks knowledge about [Bates, 1986a; Belkin et al., 1982; Blair and Maron, 1985]. Browsing is a retrieval process where the users navigate through the text database by following links from one piece of text to the next, aiming to utilize two human capabilities [Bates, 1986b]: (1) the greater ability to recognize what is wanted over being able to describe it and (2) the ability to skim or perceive at a glance. This allows users to evaluate rapidly rather large amounts of text and determine what is useful.

Chuah et al. (1994) used GOMS to model users interaction with various kinds of graphical presentation of information while they perform an airline reservation task, considering the task and graphical layouts as
presented in Casner and Larkin (1989) and Casner (1990). Casner (1990) presents four examples of graphical layouts that present the required information and examined eight different algorithms for the task. Tullis (1986) made a complete analysis of the layouts to identify and model a reasonable set of possible algorithms. From the layout algorithms, a series of CPM - GOMS operators that the model uses to solve the task are generated [Chuah et al., 1994]. In particular, the most important task operation, the visual search, was modeled with the combinations of cognitive, motor and perceptual operators [John, 1990].

A vital foundation for interactive systems designers is an understanding of the cognitive and perceptual abilities of the users [Kantowitz and Sorkin, 1983; Wickens, 1992].

The issues related to the cognitive and perceptual actions are considered as the qualitative factors, whereas the issues related to motor actions in moving and pointing a mouse are characterized as quantitative factors. The REL chart and from-to chart (travel chart or flow diagram) are used to obtain the subjective qualitative closeness relationships and objective quantitative interactions respectively, between the various pairs of components. The resulting closeness relationship values and workflow (Interaction) values are handled in the objective function of quadratic assignment model of facilities layout problem.
Koopmans and Beckman (1957) introduced the quadratic assignment problem (QAP) to model the problem of locating interactive plants of equal areas. The QAP has been applied to a wide range of applications, including urban planning, control panel layout, and wiring diagram [Bazaraa, 1975]. A recent alternative formulation of the QAP considers assigning interdepartmental distances to department pairs [Rosenblatt and Golany, 1992]. The QAP is NP - complete [Grey and Johnson, 1979], which implies that, in general, it is a hard problem to solve.

Unfortunately, there are no exerting solution procedures for a quadratic assignment problem involving more than 10 to 15 facilities [Tompkins and White, 1984]. According to Ossama and Muhittin (1993), to date, optimal solutions to general cases of the problem can only be found for problems with less than 18 departments (facilities) [Russel, D.M. and Kai-yin Gau, 1996]. Hence, heuristic procedures are used to solve quadratic assignment problems (QAP). Generally, heuristic solution procedures for the QAP can be categorized as construction and improvement procedures [Tompkins and White, 1984]. Construction procedures develop a solution "from scratch"; namely, facilities are located one at a time until all are located. A quick and dirty solution for the QAP can be obtained using a construction procedure based on the work of Conway and Maxwell (1961) and Wimmert (1959). Improvement procedure begins with all facilities located and seeks ways
to improve on the solution by interchanging or switching locations for facilities [Tompkins and White, 1984].

A classical quadratic assignment problem is formulated, handling qualitative factors and quantitative factors in the objective function to obtain the layout. Next section of the chapter presents the literature concerned with facilities layout problem based on single objective criterion.

2.2 SINGLE-OBJECTIVE FACILITIES LAYOUT PROBLEM

The facilities layout problem is concerned with the most effective arrangement of facilities in order to achieve prescribed goals such as profit or cost. It is a special case of quadratic assignment problem, and has attracted the attention of many researchers because of its practical utility and interdisciplinary importance. Historically, two basic approaches have most commonly been used to generate desirable layouts: a quantitative approach that tries to minimize the transportation cost between the facilities, and the qualitative approach which considers subjective closeness rating, to arrive at a suitable facilities layout.

2.2.1 QUALITATIVE APPROACH FOR THE FACILITIES LAYOUT PROBLEM

In the qualitative approaches, the overall subjective closeness rating between various facilities are maximized. The layout designers provide the REL charts to represent subjective closeness relationships between the facilities. These subjective closeness ratings can be used: A
(Absolutely necessary), E (Essentially important), I (Important), O (Ordinary), U (Unimportant) and X (Undesirable), to indicate the respective degrees of necessity that two given facilities be located close together. Layout designers may then assign numerical values to the ratings so that they can be handled mathematically. The numerical values assigned to the ratings have the ranking A>E>I>O>U>X.

Several qualitative approaches have been developed for the design of the layouts. Seehof and Evans (1967), Lee and Moore (1967), Muther and McPherson (1970), and Muther (1973) have developed algorithms based on qualitative criteria to obtain the layouts. The quadratic assignment formulation to minimize the distance weighted cost of closeness rating [Khare et al., 1988b] is given in equations (1) to (4).

Minimize \[ R = \sum_{i} \sum_{j} \sum_{k} \sum_{l} r_{ik} d_{jl} x_{ij} x_{kl} \] \hspace{1cm} (1)

\[ \sum_{j} x_{ij} = 1, \quad j = 1,2, \ldots, n \] \hspace{1cm} (2)

\[ \sum_{i} x_{ij} = 1, \quad i = 1,2, \ldots, n \] \hspace{1cm} (3)

\[ x_{ij} = 0 \text{ or } 1 \quad \forall i, j \] \hspace{1cm} (4)

Where, \[ x_{ij} = \begin{cases} 1, \text{ if facility } i \text{ is assigned to location } j \\ 0, \text{ otherwise} \end{cases} \]

\[ r_{ik} = \text{ Closeness relationship rating between facilities } i \text{ and } k \]
\[ d_{jl} = \text{ Distance between location } j \text{ and } l. \]
The distance weighted cost of closeness rating is considered as total numerical rating (TNR) [Sayin, 1981].

All these approaches are distinguished primarily by the scoring system used for the closeness relationship ratings and the solution procedures. For example, the ALDEP procedure presented by Seehof and Evans (1967) used the numerical values as $A = 64$, $E = 16$, $I = 4$, $O = 1$, $U = 0$ and $X = -1024$. Similarly, Lee and Moore (1967) presented the CORELAP procedure to obtain the layouts.

The major shortcomings of the qualitative approaches are as follows.

1. A method of scoring, which is based on pre assigned numerical values for different closeness ratings but does not consider material handling (flow) cost. That is, irrespective of the category of the problems, the scoring systems presented in the procedures.

2. The strong assumption is that all qualitative factors can be aggregated into one criterion. That is, the qualitative factors such as safety, noise, flexibility, aesthetics, etc. are combined and considered as one qualitative factor.
2.2.2 QUANTITATIVE APPROACH FOR THE FACILITIES LAYOUT PROBLEM

In quantitative approaches, the facilities design problems can be stated as the assignment of 'n' facilities to 'n' locations so as to minimize the cost of material handling (workflow) [Khare et al., 1988a]. The layout designers are provided with the travel charts (or flow diagrams) to represent the flows (or interactions) between the facilities.

Buffa et al. (1964), Hitchings (1973) and Armour and Buffa (1974), etc. developed improvement heuristics which improved an initial layout step - by - step, resulting in a final solution with low material handling cost. These heuristics are quantitative in nature. The cost of material handling is assumed to be an incremental linear function of distance traveled [Francis and White, 1974; Mahapatra and Bedi, 1984] for 'n' locations and 'n' facilities. The quadratic assignment formulation to minimize the material handling cost is given in equations (5) to (8).

Minimize $C = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} f_{ik} d_{ji} x_{ij} x_{kl}$  \hspace{1cm} (5)

Subject to $\sum_{i=1}^{n} x_{ij} = 1, \hspace{0.5cm} j = 1,2,\ldots,n$ \hspace{1cm} (6)

$\sum_{j=1}^{n} x_{ij} = 1, \hspace{0.5cm} i = 1,2,\ldots,n$ \hspace{1cm} (7)

$x_{ij} = 0 \text{ or } 1 \hspace{0.5cm} \forall \hspace{0.1cm} i,j$ \hspace{1cm} (8)

Where, $x_{ij} = \begin{cases} 1, & \text{If facility } i \text{ is assigned to location } j \\ 0, & \text{Otherwise} \end{cases}$

34
f_{ik} = \text{flow between facilities } i \text{ and } k

d_{ij} = \text{distance between locations } j \text{ and } l

The major limitation of quantitative approaches is that they consider only the material handling (or flow) cost that can be quantified and do not consider any qualitative factors.

The real life facilities design problem is often associated with multiple objectives such as minimization of total flow cost (TFC) and maximization of total closeness rating (TCR). Next section of the chapter presents the different multi-objective facilities layout (MOFL) design models of the various authors.

2.3 MULTI-OBJECTIVE FACILITIES LAYOUT (MOFL) PROBLEM

A multi-objective facilities design problem can be defined mathematically as the optimum assignment of 'n' facilities to 'n' locations so as to achieve the multi criteria objective [Khare et al., 1988b]. The objectives are classified into two categories, i.e., conflicting and congruent. Conflicting objectives aim at minimization of total flow cost and maximization of total closeness rating, whereas congruent objectives aim at minimization of distance weighted cost of several attributes viz., flow, closeness rating, hazardous movements, etc. Rosenblatt (1978), and Dutta and Sahu (1981, 1985) proposed an improvement type heuristic approach for solving layout problems
associated with two conflicting objectives. Sayin (1981) considered the cost as distance weighted attributes of flow and closeness rating.

The main purpose in solving the MOFL problem is to generate efficient alternatives that can be presented to the decision maker for his or her selection [Chen and Sha, 1999]. The mathematical model of QAP formulation of the MOFL problem combining both qualitative and quantitative factors into one single formation is given in equations (9) to (12).

\[
\text{Minimize } Z = \sum_{i} \sum_{j} \sum_{k} \sum_{l} A_{ijkl} x_{ij} x_{kl} \tag{9}
\]

Subject to \[\sum_{i} x_{ij} = 1, \quad j = 1, 2, \ldots, n \tag{10}\]

\[
\sum_{j} x_{ij} = 1, \quad i = 1, 2, \ldots, n \tag{11}
\]

\[
x_{ij} = 0 \text{ or } 1, \quad \forall ij \tag{12}
\]

Where, \( x_{ij} = \begin{cases} 1, & \text{If facility } i \text{ is assigned to location } j \\ 0, & \text{Otherwise.} \end{cases} \)

\( A_{ijkl} = \text{Cost of having facility } i \text{ at location } j \text{ and facility } k \text{ at location } l \text{ as defined by the application being used. That is, the cost variable representing the combination of qualitative and quantitative measures in MOFL model.} \)

Equation (10) assures that each location may contain only one facility. Equation (11) assures that a facility occupies only one location.
Many researchers have presented different cost terms and solution procedures for MOFL problem. Rosenblatt (1979), Dutta and Sahu (1982), Fortenberry and Cox (1985), Urban (1987, 1989), and Khare et al. (1988b) among others have developed multiple-criteria cost terms that consider both objectives.

Rosenblatt (1979), and Dutta and Sahu (1987) defined the cost term as follows.

\[
\alpha_2
\]

Where \( \alpha_1 + \alpha_2 = 1 \), \( \alpha_1, \alpha_2 \geq 0 \)

\[
f_{ik} d_{ij} , \text{If } i \neq k \text{ and } j \neq l
\]

\[
f_{ii} d_{ij} + c_{ij} , \text{If } i = k \text{ and } j = l
\]

\( c_{ij} = \text{Cost per unit associated directly with assigning facility } i \text{ to location } j. \)

\( d_{ij} = \text{Distance between locations } j \text{ and } l. \)

\( f_{ik} = \text{Work flow from facility } i \text{ to facility } k \)

\[
W_{ijkl} = \begin{cases} f_{ik} & \text{if locations } j \text{ and } l \text{ are neighbors.} \\ 0 & \text{otherwise.} \end{cases}
\]

\( r_{ik} = \text{Closeness relationship rating value between facilities } i \text{ and } k \)

Different solution procedures have been suggested for solving the quadratic assignment problem, either by heuristic or exact methods [Gilmore, 1962; Hillier, 1963; Hillier and Connors, 1966; Lawler, 1963;]...
and Mallette and Francis, 1972]. Rosenblatt (1979) exploited the fact that only two goals are considered in the objective function, and therefore graphical method is used to solve the problem. Dutta and Sahu (1982) presented an improvement heuristic, in which the objective function value is considered as the measure of effectiveness for the selection of new layout using a pair wise exchange routine. The scoring system used for the closeness relationship ratings is $A=6$, $E=5$, $I=4$, $O=3$, $U=2$ and $X=1$.

Fortenberry and Cox (1985) presented multiplicity QAP model and defined the cost term as follows.

$$A_{ijkl} = a_{ijkl} b_{ijkl}$$  \hspace{1cm} (16)

where, \[ a_{ijkl} = f_{ik} d_{jl} \] \hspace{1cm} \[ b_{ijkl} = r_{lk} \]

In this multiplicity model, the closeness rating values influence the cost of a layout regardless of whether facilities have common boundaries, common corners or are separated by some distance. They assigned the negative rating values for the undesirable closeness ratings so that the facilities with undesirable closeness ratings be separated. The closeness rating scoring uses the system as: $A=5$, $E=4$, $I=3$, $O=2$, $U=1$ and $X=-1$.

Urban (1987, 1989) defined the cost term as follows.

$$A_{ijkl} = d_{jl} (f_{lk} + c \cdot r_{lk})$$  \hspace{1cm} (17)
Where, $c$ is the constant weight that determines the importance of the closeness rating to the workflow. The rating system used is: $A = 4$, $E = 3$, $I = 2$, $O = 1$, $U = 0$, and $X = -1$, in order to separate the facilities with undesirable closeness ratings.

Khare et al. (1988) defined the cost term as follows.

$$A_{ijkl} = W_1 r_{ik} d_{ji} + W_2 r_{jk} d_{ij} \tag{18}$$

Where, $W_1 + W_2 = 1$, $W_1, W_2 \geq 0$.

The above models are similar in nature, and vary only in stating the relationship between the cost term $A_{ijkl}$ and the quantitative and qualitative measures. Although these models have been applied to the MOFL problem, they all have the following inadequacies.

1. Although the specific cost terms ($A_{ijkl}$) in the objective function is unique in each case, all consider only two goals or factors in the objective function. In order to judge the effectiveness of multi–goals facilities layout methods, several objectives can be used.

2. All factors may not be represented on the same scale. For example, values for workflows may range from zero to a tremendous amount, while closeness rating values may range from $-1$ to $4$. As a result, the closeness ratings would be
dominated by workflow and have little impact on the final layout.

Next section of the chapter presents the multi factor facilities layout methodology, which addresses the inadequacies of the multi-objective facilities layout models dealing with one qualitative goal (or factor) and one quantitative goal (or factor) in the objective function.

2.4 MULTI - FACTOR FACILITIES LAYOUT MODEL

Rosenblatt (1979), Dutta and Sahu (1982), Fortenberry and Cox (1985), Rosenblatt and Sinuany - Stern (1986), Urban (1987, 1989), and Khare et al. (1988b) presented quadratic assignment problem formulations of the multi-goal objectives, which handle only two goals or factors in the objective function. To judge the effectiveness of multi-goal facilities layout methods, several objectives (or factors) can be used [Harmonosky and Tothero, 1992].

Harmonosky and Tothero (1992) presented the formulation for the multi-factor facilities layout problem, handling a number of qualitative and quantitative factors in the same manner in the objective function. That is, all qualitative as well as quantitative factors are assigned with equal weights so that the sum of all weights is unity. Several different qualitative relationship charts may be used to represent different kinds of relationships. For example, the qualitative relationships desired by
different managers may also be considered separately. The use of common utilities by facilities may be considered separately from the environmental effects between facilities. It is also possible to consider several quantitative aspects. For example, flow may be separated by material size or product line.

In real cases, all the factors may not be represented on the same scale and measurement units used for objective may be incomparable [Chen and Sha, 1999]. To overcome this problem, Harmonosky and Tohero (1992) proposed a methodology that normalizes all factors before combining them.

The methodology begins by combining individual factors, qualitative and quantitative alike into a single composite factor for the objective function. First, all qualitative factors are quantified so that they may be handled mathematically. Second, all factors are normalized so that each will have an equivalent effect on the layout. To normalize a factor, each relationship value is divided by the sum of all relationship values for that factor as given in equation (19).

$$T_{ikm} = \frac{S_{ikm}}{\sum_{i} \sum_{k} S_{ikm}}$$  \hspace{1cm} (19)

Where, $S_{ikm}$=relationship (workflow) value between facilities $i$ and $k$ for factor $m$.

$T_{ikm}$ = normalized relationship (workflow) value between facilities $i$
and \( k \) for factor \( m \).

The result of this division is that sum of all relationship values for any factor will be one.

Next, all values are multiplied by weights \( (a_m) \) for each factor \( m \). Finally, all factors are combined into one composite factor \( (A_{ik}) \) as given in equation (20).

\[
A_{ik} = \sum_{l} \sum_{k} \sum_{m} a_m
\]

Where, \( t = \) number of factors.

Then, the objective function \( (A_{ijkl}) \) value is obtained as given in equation (21).

\[
A_{ijkl} = A_{ik} d_{jl}
\]

Where, \( d_{jl} = \) distance between locations \( j \) and \( l \).

In the Harmonosky and Tothero (1992) model, the closeness relationship ratings of qualitative factors and workflows of quantitative factors are represented on the same scales. But, all the qualitative and quantitative factors are handled in the same manner. That is, the sum of weights assigned to all qualitative and quantitative factors is unity. Moreover, all the individual qualitative factors as well as the quantitative factors are assigned with equal weights. As a result, each of the
qualitative factors as well as the quantitative factors will have an equivalent effect on the final layout.

A single - objective facilities layout models are classified into two categories. One is handling qualitative closeness relationship values of one qualitative factor, and the other is, handling quantitative workflow values of one quantitative factor in the objective function. In the MOFL models, all the qualitative factors (objectives or goals) are aggregated into one qualitative factor (objective or goal) and quantitative workflow values of one quantitative factor (objective or goal) are handled in the objective function. They may be conflicting objectives or congruent objectives. The multi - factor facilities layout model, handles all the individual qualitative and quantitative factors in the same manner in the objective function. The classification of the facilities layout models is shown in Fig. 2.1 as follows.

Facilities Layout Models

<table>
<thead>
<tr>
<th>Single objective</th>
<th>Multi objective</th>
<th>Multi factor facilities layout Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualitative Objectives</td>
<td>Quantitative objectives</td>
<td>Conflicting objectives</td>
</tr>
</tbody>
</table>

Fig. 2.1: Classification of Facilities Layout Models
Once the composite factor has been obtained, then the problem is solved as a single factor problem either by construction procedure or improvement procedures.

Next section of the chapter presents the solution procedures for the facilities layout problems.

2.5 FACILITIES LAYOUT PROCEDURES

The solution procedures for quadratic assignment problem (QAP) fall into two categories, such as optimal and heuristic procedures. Gilmore (1962), Lawler (1963), and Gavett and Plyter (1966) offered exact solution procedures using branch and bound techniques. But owing to computational inefficiency of the optimal procedures for a large number of facilities, heuristic algorithm models have drawn researcher's attention [Houshyar and Bringelson, 1998]. Because the applicability and performance of one heuristic over another can be subjective and based on many factors [Tompkins and White, 1984]. Heuristic procedures are of either construction type or improvement type.

Construction procedures build layout from scratch by making successive assignments of facilities to sites, whereas improvement procedures begin with an initial assignment of facilities to locations and attempt to iteratively improve on it, such as CRAFT by Armour and Buffa (1963).
2.5.1 CONSTRUCTION PROCEDURES

The construction type solution procedures, usually provide a layout based on closeness rating between facilities [Khare, et al., 1988]. Delta hedron (DA) approach, MATCH, SPIRAL, SHAPE, NLT, QLAARP are construction algorithms, among which DA approach, MATCH and SPIRAL are adjacency-based algorithms, and the others are distance based algorithms.

DELTA HEDRON APPROACH (DA): The DA of Foulds and Robinson (1978) maintains a planar graph which allows for an easier transportation to block layout. It proceeds by determining the sequence that nodes will enter the graph. At any stage, a node is entered into the center of the face (a triangle formed by three nodes) in the graph that will maximize the adjacency benefits with the other facilities in the face. Al-Hakim (1991), Boswell (1992), and Leung (1992) developed algorithms to improve on DA’s performance. The DA has also been modified to consider a continuous relaxation of the adjacency decision variables using a shortest path approach [Griffin and Foulds, 1987].

MATCH : It is an Interactive construction – type approach that utilizes a discrete representation and integer programming to solve a b-matching problem [Montreuil, Ratliff, and Goetschalckx, 1987]. This algorithm attempts to find a matching that maximizes the adjacency score while
satisfying the lower and upper bound on the number of matches with each facility and the total number of times of facility must be matched with all other facilities, and computed adjacency scores based on number of adjacent segments.

SPIRAL: It uses the concept of “relationship tuples” to construct a adjacency graph, and then a block layout is developed in a row – and – column structure [Goetschalckx, 1992]. The tuples quantify the relationship between one facility and other facilities.

SHAPE: The facility selection to place at the center of the layout is based on a ranking, which is based on the flow of each facility and a user – defined critical flow value. Subsequent facilities placement is based on the objective function value dependent on rectilinear distances between centroids placed on each of the layout’s four sides [Hassan, Hugg and Smith, 1986]. This algorithm is easy to implement, but the location shape is controlled by the objective function, the shape of departments (locations) may deteriorate towards the end.

N L T: The non-linear optimization layout technique (NLT) is based on non-linear programming techniques and utilizes Euclidean distances between facilities centroids [Van Camp, Carter and Vannelts, 1991]. In the NLT model with constraints, such as locations (departments) cannot overlap, cannot be located outside the location and cannot be assigned area less than required, is transformed to an unconstrained form by an exterior point quadratic penalty function method. With a three-stage
approach, successively more difficult problems are solved using the solution from the previous stage as an initial solution point. The departments (locations) shapes are all rectangular.

**QLAARP**: Qualitative Layout Analysis using Automated Recognition of Patterns (QLAARP) uses qualitative layout anomalies (QLA's) [Benerjee et al., 1992] to set binary variables in Montreuil’s mixed integer programming [Montreuil’s, 1990]. This algorithm heuristically uses context-based information to reduce the solution tree.

### 2.5.2 IMPROVEMENT PROCEDURES

Improvement procedures begin with an initial assignment of facilities to locations and attempt to improve an initial layout iteratively, step-by-step resulting in a final solution with low material handling cost. CRAFT, multi goal heuristics of Rosenblatt (1979), Dutta and Sahu (1982), and Fortenberry and Cox (1985) are the improvement procedures for the facilities layout problems.

**CRAFT**: Computerized Relative Allocation of Facilities Technique (CRAFT) [Armour and Buffa, 1963] begins with an initial layout. It then performs two way or three way exchanges of the centroids of non-fixed departments that are also equal in area or adjacent in the current layout. For each exchange, CRAFT computes an estimated reduction in cost and it chooses the exchange with the largest estimated reduction.
It then exchanges the departments exactly and continues until there exists no estimated reduction due to two way or three way exchanges. Constraining the feasible department exchanges to those departments that are adjacent or equal in area is likely to affect the quality of the solution, but it is necessary due to its exchange procedure.

**MULTI-GOAL HEURISTICS:** Rosenblatt (1979) developed a multi-goal heuristic, which results in a discrete efficient frontier set \( E \) consisting efficient layouts. The weights \( \alpha_1 \) and \( \alpha_2 \) or range of weights for the goals are specified to generate the best layout. This heuristic begins with generating random layout, and computing the total closeness rating \( R_j \) and total material handling cost \( C_j \). The point \( (R_j, C_j) \) is compared with the existing point \( (R_i, C_i) \) in the set. If \( R_j \leq R_i \) and \( C_j \geq C_i \), then \( (R_j, C_j) \) can be eliminated. Otherwise, all points within the set are compared with \( (R_j, C_j) \), and the dominated original points within the set are discarded. The values of \( \alpha_1 \) and \( \alpha_2 \) are specified for all the efficient points of different layouts to generate the best solution, which results in minimum \( (\alpha_2 C - \alpha_1 R) \). If \( \alpha_1 \) and \( \alpha_2 \) are not specified, the sensitivity analysis with respect to \( \alpha_1 \) and \( \alpha_2 \) may be performed to evaluate the best layout.

Dutta and Sahu (1982) proposed an improvement heuristic for multi-objective facilities layout problem, which requires an initial layout to minimize the total material handling cost and maximize the closeness
rating with predefined weights assigned to them. This pair-wise
exchange heuristic starts with computing the measure of effectiveness;
\[ C' = W_2 C - W_1 R, \]
where \( C \) is the total material handling cost and \( R \) is the
total closeness relationships scores and \( W_2 \) and \( W_1 \) are the weights
assigned to them. The new \( C' \) is computed after every exchange of the
pair of facilities, and compared with the existing \( C' \) to facilitate the
exchange of the facilities. This process is continued for all the pairs of
facilities, and the best layout with minimum \( C' \) is obtained.

Fortenberry and Cox (1985) proposed another improvement
heuristic for the multi - goal facilities layout problem, which improves
the initial layout in a multi pass step-by-step pair wise exchange. This
heuristic is similar to the ones used by Dutta and Sahu (1982), and
Francis and White (1974). This heuristic is involved with computation of
low total cost for the layout (TC). For every exchange of the pair of
facilities the new total cost is computed and compared with low total
cost. If the new total cost is less than low total cost, then it is set to low
total cost. The process is repeated for all the exchanges of facilities and
the low total cost for the best layout is obtained.

The layout procedures, which combine construction and
improvement heuristics to obtain the efficient layouts are also
developed. For example, ALDEP of Seehof and Evans (1967) uses both
construction and improvement procedure [Houshyar and Bringelson, 1998].

Later, Harmonosky and Tothero (1992) developed a heuristic which combines a construction and improvement procedure to quickly develop a good layout for the multi-factor facilities layout problem.

2.5.3 PROCEDURE FOR THE MULTI-FACTOR FACILITIES LAYOUT PROBLEM

Harmonosky and Tothero (1992) presented a layout procedure, in which a construction method is employed to obtain a good starting initial layout for an improvement method based on pair-wise exchanges.

CONSTRUCTION HEURISTIC: It involves selecting the pair of facilities with the highest composite relationship to place it near the center of the layout, and computing the cost incurred by placing this pair in the layout. The facility with highest relationship value with one of the facilities in the layout is selected to place in the location that results in the lowest additional cost. The procedure is repeated for all the pairs of facilities and total cost is computed.

The layout generated in the construction procedure is used as a starting layout for the improvement procedures, which improves the solution based on pair-wise exchanges.
IMPROVEMENT PROCEDURE: In the improvement procedure, the facilities trade locations in an attempt to lower the cost for the layout based on pair-wise exchanges. Generally, the layout obtained in the construction heuristic is used as a starting layout for the improvement procedure. Harmonosky and Tothero (1992) presented a pair-wise exchange heuristic, which allows more locations to be available than facilities, and hence, some locations will be left vacant after the layout is completed.

The pair-wise exchange heuristic begins with computation of the total cost for an initial layout. Initially, the maximum cost of saving for exchanging two facilities as well as the maximum cost of saving for moving a facility to an unoccupied location are set to zero. Then the pair of facilities, which results in maximum cost of saving, is selected for exchange and the cost of saving is computed. The cost of saving is compared with the maximum cost of saving for exchange. If the cost of saving is greater than the maximum cost of saving for exchange, then the maximum cost of saving is set to the cost of saving for exchange. If the shape of the layout is allowed to change, then the facility is selected to move into an unoccupied location that results in maximum cost of savings for a move and the cost of saving is computed. If the cost of saving is greater than the maximum cost of saving for a move, then the maximum cost of saving is set to cost of saving for a move. If the cost...
of saving for a move is greater than the cost of saving for an exchange, then a facility is moved to a location, and the total cost for a new layout is computed. Otherwise, an exchange is made and total cost for a new layout is computed. The procedure is repeated until there is no further improvement possible in the layout.

The improvement heuristic of Harmonosky and Toothero (1992) layout procedure allows more locations to be available than facilities, and hence some locations will be left vacant after the layout is completed by allowing the change in shape of the layout.

Though the composite criteria values are included with workflows (interactions) and closeness relationships values between the facilities, the pair of facilities with highest composite criterion value is selected in construction heuristic of the multi-factor facility layout procedure to generate an initial layout.

The quadratic assignment models for the facilities layout problem and the solution procedures are found to be equally effective in the design of human-computer interactive systems. The problem of interactive systems design varies from location and layout of facilities in a manufacturing plant to the design of the textual and graphical user interface components layout in the human-computer interface. It has been found that there is a sort of one-to-one relationship between
arrangement of manufacturing facilities with that of the textual and graphical user interface components (icons). In the problem of interface design, the facility can be a text or graphical user interface component (icon), and the issues related to the cognitive and perceptual actions namely, fatigue, attention, monotony, boredom, etc. might be considered as the qualitative factors, whereas the issues related to motor actions in moving and pointing a mouse such as, frequency of use, interaction style, step-by-step work, all-at-once work, etc may be characterized as the quantitative factors.

Next section of the chapter presents the qualitative and quantitative factors concerned with human-computer interface design.

2.6 QUALITATIVE AND QUANTITATIVE FACTORS OF THE HUMAN - COMPUTER INTERFACE

The qualitative and quantitative factors concerned with the various types of user viz., novice, intermittent, expert for the layout design of the textual and graphic user interface components are presented as follows.

2.6.1 QUALITATIVE FACTORS

The issues related to the cognitive and perceptual, actions, such as familiarity, interface type, instruction type, fatigue, attention, monotony, boredom, anxiety, fear, etc. are considered as qualitative factors for the layout design of the textual and graphical user interface components.
A task that involves a communication with natural (real-world) objects is classified as a familiar task, and a task that involving objects and actions not consistent with a typical person's experience is considered as a less familiar task. The more familiar a task, the more likely automaticity is to occur because a routine for performing the task can be retrieved from memory [Logan, 1988a; Logan and Klepp, 1992]. For a novice user, using a poorly designed HCI system may require heavy cognitive effort. On the other hand, for an expert user, the same HCI system becomes routine, and uses the system without much cognitive effort. For familiar tasks, people think about what to do rather than the mechanics of how to do it. This results in automatic processing for all levels beneath the goal [Logan, 1992]. Familiar activities can be identified at a higher (goal-oriented) level, and thus facilitate automatic processing, whereas unfamiliar activities are identified at a lower level, limiting automatic processing. Sometimes, familiar tasks may mismatch with the type of task instruction, i.e., specific versus interpretive, and processing will be inefficient.

Interface type is another between-subjects factor [Lim et al., 1996]. The interfaces may be menu-based interfaces and the direct-manipulation interfaces. Both the interfaces have the same set of icons appearing in the same position on the computer screen. Actions in the direct manipulation interface are executed by dragging two objects
(icons) together. For the menu based interface, actions are executed by selecting one icon from each menu. The total time required to complete a task in a direct-manipulation interface will be less than that in a menu interface, since less non-motor time will be required to complete a task in a direct-manipulation interface than in a menu-based interface.

Within each interface type, there may be two types of instructions viz., specific instructions and interpretive instructions. Both the sets of instructions require the same physical task to be performed and produce the same output. The specific instructions prescribe an action plan that tells the user how to complete the task, and the interpretive instructions specify what needs to be done, but not how. An interpretive instruction requires user to begin at the goal level, translate the goal into an intention, and then formulate an action plan for executing that goal, using the mouse device. In contrast, a specific instruction provides the user directly with an action plan in a step-by-step manner, which needs to be executed. For interpretive instructions of a less-familiar task, users need to engage in a multistage process which begins by translating the interpretive instruction “retrieve incoming memo”, which describes what to do, into an action plan, that describes how to do it, followed by execution of the action plan. Hence, more non-motor time is required to complete a less-familiar task than a familiar task. On the other hand, for specific instructions of a less-familiar task, less non-
motor time is required to complete a task than a familiar task [Lim et al., 1996].

Fatigue is related to feeling of tiredness in performing the task. For expert users, the rate of fatigue is less compared to the novice users. For the user of less familiar tasks, fatigue is developed at fast rate than the users of familiar tasks. The interface type as well as instruction type affect the fatigue of the users.

Monotony is related to the wearisome in absence of variety, which results in boredom. The users of familiar tasks are subjected to monotony and boredom, and hence result in fatigue. Monotony and boredom is also developed in novice users of un-familiar tasks, which may turn to fatigue. The fatigue resulted from the un-familiar tasks and familiar tasks of the novice users cause an anxiety and fear in the users, which affects the performance of the users.

The subjective impressions or reactions regarding the issues of the cognitive and perceptual activities between the pairs of components are collected in the form of rating scales of ranking [Shneiderman, 2000].

2.6.2 QUANTITATIVE FACTORS

The quantitative factors related to motor actions in moving and pointing a mouse for the textual and graphical user interface
components are characterized as frequency-of-use, interaction style, pace of interaction, step-by-step work, all-at-once work, and number of errors per hour.

If users can accomplish frequent tasks by moving through a display in a top-to-bottom pattern, then faster performance is likely compared to that with a layout that requires numerous jumps around widely separated parts of the display. Designers specify the sequences of selection that users make and the frequencies for each sequence. When the frequently used icons, buttons or boxes are clustered, then an optimal layout that minimizes visual scanning with increased accuracy and reduced interaction between the pairs of components can be produced.

The interaction style is more appropriate for knowledgeable intermittent users or frequent users. Based on the interaction style, the number of interactions vary. As the frequency of use increases, so do the users desire to reduce the number of interactions and to increase the pace of interaction. A good layout design of the user interface components increases the pace of interaction. People may have different preferences for interaction style and pace of interaction, which is definitely based on the dense or sparse arrangement of components.
During the initial learning stage a task is carried out step-by-step, or controlled manner, which requires the sparse arrangements of components so that the user can be clear in his choice of selecting the components to perform the task. It may increase the number of interaction between the pairs of components.

At some point an individual will have learned enough to be able to retrieve solutions directly from memory rather than formulating the procedure to solve the problem. That is, people learn specific solutions through repetitive practice with specific problems and hence automaticity occurs. Also, at this point, they should be developing the ability to generalize solutions to similar problems within that domain [Logan, 1988a], and the procedure results in all-at-once to perform the task. In order to perform the task in all-at-once requires the dense arrangements of components so that the user can perform the task fast in less time with minimum number of interactions.

The use of the less-familiar icons led to a higher number of errors [Benbasat and Todd, 1993]. The number of errors of novice, intermittent, and expert users vary for interaction styles, pace of interaction, step-by-step, and all-at-once work, and so on. A good layout design of user interface components suitable for novice, intermittent, and expert users may reduce the number of errors. The
increase in number of errors results in increasing the desired number of interactions between the components.

The values of the quantitative interactions between the various pairs of components of these factors may be recorded in the travel chart (from- to chart or flow diagram). There is evidence that the issues related to cognitive and perceptual actions as well as the issues related to the motor actions tend to overlap [Grey et al., 1993; John, 1990]. The numerical values assigned to closeness ratings of qualitative factors as well as the numerical values of the quantitative factors for the various types of users are included in the objective function of quadratic assignment model for the layout design of graphical and textual user interface components.

SUMMARY

This chapter presents the literature concerned with the theories and models used in the design of human – computer interaction (HCI), facilities layout models solution procedures of the facilities layout problem, and the qualitative and quantitative factors of user interface components layout problem. The various human – computer interaction (HCI) models with their merits and demerits are presented in section 2.1. The single objective facilities layout models, which consider either qualitative criteria or quantitative criteria for the facilities layout problems are presented in Section 2.2. Section 2.3 discusses the
various multi-objective facilities layout (MOFL) models, which handle one qualitative factor and one quantitative factor in the objective function. Section 2.4 presents the multi-factor facilities layout model, which handles a number of qualitative and quantitative factors in the same manner in the objective function. Section 2.5 presents the various solution procedures of facilities layout problem, based on qualitative criterion, quantitative criterion, and qualitative and quantitative criteria. Section 2.6 presents the various qualitative and quantitative factors that affect the layout design of human-computer interface components.