

# AN INTELLIGENT APPROACH TO DESIGN TASKS

Thesis Submitted to  
The University of Hyderabad  
for the award of the degree of  
Doctor of Philosophy

by

P. NAGA PRASAD



Department of Computer and Information Sciences  
School of Mathematics and Computer/Information Sciences  
University of Hyderabad  
Hyderabad - 500 134  
INDIA

August 1993

## CERTIFICATE

This is to certify that the thesis entitled '**An Intelligent Approach To Design Tasks**' being submitted by **P. Naga Prasad** for the award of the degree of Doctor of Philosophy in Computer Science to the University of Hyderabad, is a record of bonafide and original work carried out by him under my supervision.

The matter embodied in this thesis has not been submitted to any other University or Institute for award of any degree.



**Dr. A. Sivasankara Reddy**  
**Supervisor**



**Head**  
**Department of Computer/Information Sciences**



**Dean**  
**School of Mathematics and Computer/Information Sciences**  
**University of Hyderabad**  
**Hyderabad**

## DECLARATION

I, P. Naga Prasad, hereby declare that the research work embodied in the present thesis entitled '**AN INTELLIGENT APPROACH TO DESIGN TASKS**' carried out by me under the supervision of Dr. A. Sivasankara Reddy is my original work. I assert that this thesis has not been submitted or published either in parts or in full by any university for the award of any degree either in India or abroad.



P. Naga Prasad

DEDICATED TO MY PARENTS

*P. Madhusudana Rao*

*P. Bharathi Devi*

## ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude to *Dr. A.S. Reddy*, my research supervisor, without whose invaluable guidance and support, this thesis would not have been possible. His guidance at all stages of the project is gratefully acknowledged.

I am indebted to *Dr. P.G.Reddy*, Senior Professor, *Dr. B.E. Prasad*, Professor and Head, Department of Computer/Information Sciences and *Dr. V. Khannan*, Professor and Dean, School of Mathematics & Computer Information/ Sciences, University of Hyderabad and *Dr. S. Ramaswainy Reddy*, Professor and Principal, C.B.I.T., Hyderabad for their encouragement.

I acknowledge *Dr. B.R. Sastry*, Professor and Head, Department of Computer Science and Engineering, C.B.I.T., Hyderabad, for his constant persuasion and needful support to complete this project. Also, I acknowledge *Dr.E. Baiagurusaujy*, Professor and Director, National Centre for Expert Systems, Hyderabad and *Dr.B. Susheela*, Executive Director, Institute of Chartered Computer Professionals of India, Hyderabad for being constant sources of inspiration.

I wish to appreciate the help provided by *Sri P. KanaJcaraju*, Scientist, MIS, SPROB, Indian Space Research Organisation, Sriharikota at various phases of the present project.

I am grateful to *Sri A. Nagaraja Sekhar*, Senior faculty of Electrical Engineering Department, C.B.I.T., and *SriB. Srinivas*, Engineer, SPROB, ISRO, Sriharikota for providing their expert knowledge in Electrical and Mechanical Engineering respectively.

I would like to thank my friends *Dr. G.V. Subba Rao*, *Dr. M.V. Raman a Murthy*, *Sri U. V. Ramanaiah* and *colleagues* of the Department of Computer Science and Engineering, C.B.I.T.,who helped me during the research work.

I am thankful to *Sri H.R.L. Narasimhan*, Osmania University., for his diligence in preparing the graphs and figures.

Finally, my special thanks are due to *Sri P. Srinivasa Rao*.

## TABLE OF CONTENTS

Abstract		v
<b>Chapter 1</b>	<b>Foundations of Generic Design Tasks</b>	
1.1	Introduction	1
1.2	History of Expert Systems for Design Tasks	2
1.3	Understanding Design Activity	3
1.3.1	Definition of Design	4
1.3.2	Design as an Intelligent Activity	7
1.3.3	CAD and I-CAD	9
1.4	Generic Nature of Design Activity	12
1.4.1	Classifications of Design Tasks	13
1.4.2	Design is an Exploration Based Activity	14
1.4.3	Search is the Process of Design	20
1.5	Summary	25
<b>Chapter 2</b>	<b>Modeling Design Process - representative Design Process Model(rDPM)</b>	
2.1	Introduction	27
2.2	Modeling Design Task	28
2.3	Towards representative Design Process Model(rDPM)	31
2.3.1	Knowledge in Design Activity	32
2.3.2	Constraints in Design Activity	37
2.3.3	The Role of Operations	38
2.4	Theory of Design Process	38
2.4.1	Design Process Theory(DPT)	39
2.4.1.1	The Origins of $F^e$ and $F^d$	45
2.4.1.2	Criteria of Object Decomposition	50
2.5	representative Design Process Model(rDPM)	58
2.5.1	Description of rDPM	58

2.6	rDPM and Knowledge Organization	63
2.6.1	Knowledge at select Phase	63
2.6.2	Knowledge at decomposition Phase	64
2.6.3	Knowledge at integrate Phase	65
2.6.4	Knowledge at scrutiny Phase	65
2.6.5	Knowledge at alter Phase	66
2.7	Summary	67

### **Chapter 3 cognitive Design Process Model(cDPM)**

3.1	Introduction	69
3.2	Human Brain Vs Computer	70
3.2.1	The Architecture of Human Brain: An out Line	72
3.2.2	The Functional Architecture of Computer: An out Line	73
3.2.3	Limitations of Brain and Computer	76
3.3	Cognition and Design	78
3.3.1	Protocols in Design of Various Domains	80
3.3.1.1	Design of Process Heat Exchanger	80
3.3.1.2	Design in Structural Engineering	81
3.3.1.3	Electrical Circuit Design	82
3.3.2	The Role of Analysis and Synthesis	84
3.3.2.1	Relations between Analysis and Synthesis	87
3.3.3	Cognitive Modeling of Design Process: Previous Work	87
3.3.4	cognitive Design Process Model(cDPM)	100
3.4	Summary	105

### **Chapter 4 Knowledge Engineering in generic Intelligent CAD System**

4.1	Introduction	108
4.2	Modeling Design at System Level and Life Cycle Model	1 10
4.2.1	Design Planning	1 1 1
4.2.2	Design Scheduling	1 1 1
4.2.3	Design Implementation	112

4.3	Analysis of Design process in the Domain of DC Power Supplies	1 15
4.3.1	Design of The Transformer	116
4.3.2	Design of The Filter	123
4.3.3	Design of The Rectifier	127
4.4.	Expert System Development	130
4.4.1	Expert System in Design	130
4.4.2	An Experimental Working-System Development for DC Power Supplies Design	131
4.5	Further Design Examples	
4.5.1	Screw Jack, an example of mechanical engineering Design	1 38
4.5.2	Thermal Design of 1,2 shell & tube Heat Exchanger	141
4.6	Analogy between Design of DC Power Supplies, Screw Jack & 1,2 shell & tube Heat Exchangers	146
4.7	Summary	148

## **Chapter 5 Conclusions and Future Directions**

5.1	Summary and Conclusions	151
5.2	Future Directions	156

## **Appendices**

A	Functions, Constraints and Components of DC Power Supplies	158
B	Design of Screw Jack System	161
C	Design of 1,2 shell & tube Heat Exchangers	173

<b>References</b>		<b>185</b>
-------------------	--	------------

# ABSTRACT

*Understanding design processes are particularly significant in light of their complexity, inconsistency and intuitive behaviors involved and hence became a subject to AI community. It is relatively recent that a global initiative was commenced. Studies are undertaken to understand design process as a multidimensional subject: at representative, cognitive and system levels. It is under this global objective that this thesis is attempted.*

*As a prelude, the present study postulated, what is called generic **Design Process Theory (DPT)**. Here, various propositions are proposed mainly to form a basis for understanding design in addition to model the design process. Unlike the earlier models, it is argued that a further splitting of function set into two sets: essential functions and desired functions, dramatically improves the understanding of design process concepts. In addition, it is found that it forms a basis for innovative or creative designs.*

*The ideology behind the present thesis' frame work is to model design process upon understanding human epistemology. Three models need to be suggested, at representative level, at cognitive level and at system level. These models are evaluated by analyzing illustrative design tasks from different domains.*

*Design is identified as processed at two planes: the knowledge plane and the parametric plane. At knowledge plane, design knowledge is manipulated in such a way that it facilitates designer to take decisions, judgments, choose between options besides driving the process to progress to next step. At parametric plane, the data is substituted and object description data is arrived at.*

*The study proposes that the availability of the primitive form of the object under consideration for design is necessary and sufficient condition. The primitive form is imagined in the present study as an ensemble of different subcomponents having their own boundaries still connected to adjoint subcomponents. The actual*

*design of every subcomponent with required improvements followed by synthesis is an evolutionary process. Hence, design is proposed as an evolutionary synthesis process demanding judgment and decision making.*

*Based on DPT, an attempt was made to model the design process at representative level. **rDPM(representative Design Process Model)** proposes a paradigm **select-decompose-integrate-scrutiny-alter** as a representative of design process.*

*As a matter of fact, understanding is not complete, if it is not understood at cognitive level. The present study tries to apprehend the human epistemological activity while designing and model which is called **cDPM(cognitive Design Process Model)**. **cDPM** suggests that the three abductive, inductive and deductive inferences are used recursively at various phases of design. Besides, **cDPM** identified that design synthesis predominates abduction and induction reasoning mechanisms.*

*At the system level, a design life cycle model is suggested. It is identified that design process is at three layers: **planning, scheduling** and implementation. An attempt was made to assimilate this system level model to **rDPM** and **cDPM**.*

*Finally, the theory proposed is validated on an experimental basis by constructing an expert working-system **X-DAPS(Expert Design Assistant for Power Supplies)**. DC power supplies are taken as an example because it is confessed that the example is sufficiently general in the context of expert system architecture. In addition to this, two more examples from different domains, the mechanical and chemical engineering were analyzed. The analysis of the task of Screw Jack design which is a case of higher complexity and the task of thermal design of Heat Exchanger which is a case of higher inconsistency infer two issues: 1) design process is generic in nature independent of domain and 2) the features of **X-DAPS** are sufficiently general to be suitable for any design.*

## Foundations of Generic Design Tasks

### 1.1 Introduction

Artificial Intelligence, as the name implies is a subject of emulation of human activities. As a science, Artificial Intelligence has matured from past fifty years. Artificial Intelligence(AI) gained its significance due to the conviction that human cognitive skills **can** be explained and formalized. AI researchers began to consider the feasibility of **automating the process of intelligent behavior** and in this epoch, AI is spearheading into almost all scientific fronts. Emulating human brain in solving problems is a subclass of AI related tasks and is called Expert Systems. Commercial success of various Expert Systems for various human specific **applications(Hayes-Roth, 1983)** such as mineral survey(PROSPECTOR, **Duda et al, 1979**),Computer Configuration(RI, **McDermott, 1981**), Chemical structure **elucidation(DENDRAL, Lindsay et al, 1980)**, diagnosis complex test **cases(CADUCEUS, Pople, Miller and Mayers, 1975,1981,1982)**, pulmonary function disease(PUFF, **Feigenbaum 1977**) to name a few, gave a boost to the researchers to further explore newer and newer areas. One/<sup>example</sup>of such application fronts is tasks related to design.

It is well known that Expert Systems essentially need reasoning **mechanism** and knowledge to reason. Often, it is domain knowledge whose availability becomes critical for the success of Expert Systems. Definitely, design tasks too **axe** knowl-

edge intensive, nevertheless, availability of knowledge is a major constraint. Many times, there are no tractable algorithmic solutions and often these problems resist precise description and rigorous analysis. In the view of **Hayes-Roth, Waterman & Lenat(1983)**, expert's knowledge per se seems both necessary and nearly sufficient for developing an expert system. Though this observation is **empirical**, it is agreeably true in case of expert systems for design tasks.

## **1.2 History of Expert Systems for Design Tasks**

Understanding design tasks are particularly significant in the light of their **complexity**, inconsistency and intuitive behaviors involved and hence became a subject to AI Community. The history of AI applications **in** design activities traces back over three decades. Perhaps **Godwin's(1958)** program on design of large electric motors is the first AI based Expert System on design. Since then, its progress was rather sluggish till 80's and it is recently that most of the work on design problems are published by various workers worldwide. In fact, history of expert systems for design tasks are categorized into two phases. The initial phase is characterized by the orientation of researchers inclined to develop workable programs on different application areas. To name a few are **SEMCI(Neven & Harven, 1986)** an expert system for civil engineering design, **EDISON(Dyes et al, 1986)** an engineering design invention system operating naively, **HIRISE(Maher & Fenves., 1984)** an expert system for structural design of High Rise Buildings and **ALL-RISE(Sriram & Fenves, 1986)** an expert system for structural design, **SACON(Bennett et al, 1978)**, **RI** or **XCON(McDermott, 1980)**, **LIFT(Bremdal & Kristiansen, 1986)** an expert system for Marine system design, **IDRILL(Lu, Blattner & Lindem., 1986)** an intelligent drilling station design, **AIR-CYL(Brown,**

**1985) an expert system for designing air cylinders by giving specific requirements, expert system for V-belt drives(Dixon & Simmons, 1984), expert system for mechanical design(Brown & Chandrasekharan, 1983).** Rychener's review reveals comprehensively the amount of work done upto1985 on engineering design using Artificial Intelligence(Rychener, 1985). The list is not exhaustive rather indicative of expert system development for spectrum of applications.

The later phase, which is confessed it as still continuing is the scientific and epistemological understanding of design as a task. Design tasks are tried to be classified, modeled upon treating it as a generic activity. At this context, people are worried about their degree of understanding design activity. **David steier(Steier, 1990)** on reviewing the proceedings of the workshop on January, 1990 at EDRC of Carnegie-Mellon university on design systems, expressed a question on design tasks that calls for a serious thought:what **do we know** ? The reason is **not hard to conceive - Design is intrinsically complex activity and the theories of AI in this respect are too weak to support, if not impossible.**

### **1.3 Understanding Design Activity**

Design is well known as a creative human activity and is a process of providing form from a formless description. It was identified in the present study that there are four reasons why design tasks are difficult to automate:

1. *The judgments involved are subjective in nature.*
2. *Grasping relations in associations among various components is contextual and human specific.*

3. *Design needs developed techniques of pattern matching from past experience and*
4. *Computers are best for deductive logic, whereas, design demands larger proportions of abductive and inductive logics.*

Understanding design as an activity invariably throws some light on constructing expert systems. There are three categories of design science according to Warfield (**Warfield, 1990**): **(a) Specific design science (b) Generic design science and (c) General design science** . Specific design as **the** name implies is restricted to a single discipline and scarcely associated with more than one discipline, On the other hand, Generic design science deals with the common arguments such as characteristics, attributes, phenomena and conditions among all design situations irrespective of the kind of application. General design science comprises of both kinds of design science and hence, a complete design science. In his opinion, this concept remains unrealized. The present study is aimed towards Warfield's generic design science. What it does mean by understanding design tasks is the concept formation of generic design activity.

The idea behind this chapter is two fold: besides reviewing the present state-of-the-art technology of understanding design, also<sup>it</sup> identifies the necessity for an unified omnimodel for design tasks that represents design problems(generic in nature).

### **1.3.1 Definition of Design**

Design, as Oxford dictionary puts it, is a preliminary plan or sketch for making or production of building, machine, garment etc., and the

**art of producing these. Nevertheless, many a number of researchers tried to define design in their own context. Among which a few are mentioned here.**

**1. Mastow, 1985:**

*Design can be viewed as constructing an artifact of structure that (1) satisfies a given functional specification (2) conforms to the limitations of the resources and (3) satisfies implicit and explicit design criteria on the form of the artifact.*

**2. Waterman, 1985 :**

*Design is configuring objects under constraints.*

**3. Sriram & Maher, 1986 :**

*Design is perhaps the toughest problem solving activity that engineers perform and the process of design involves integrating and satisfying constraints from a number of resources. Functional specification, the limitations of the resources and the design criteria can all be treated as constraints on the final product.*

**4. Frost, 1986 :**

*Design involves the specification of a system or object such that it satisfies some given set of requirements and can be built using some given set of resources.*

**5. Simmons & Dixon, 1986 :**

*Design is a static description of the object. It does not describe the exact sequence of operations by which it is to be manufactured.*

**6. Boyle, 1989 :**

*Design is the realization of an artifact having some design function(s).*

**7. Gero & Rosenman, 1990 :**

*The aim of the design is to produce a description of an artifact which will exhibit the necessary attributes to carry out a given function. Thus design is a process of providing form from a formless description.*

**8. Smithers et al, 1990 :**

*The purpose of a design task is to devise a specification of how the environment could be changed and the actions necessary to achieve the desired alteration.*

**9. Chandrasekharan, 1990 :**

*The design problem is specified by (1) a set of functions(explicit & implicit) to be delivered by an artifact and a set of constraints to be satisfied and (2) a technology, that is, repertoire of components assumed to be available and a vocabulary of relations between the components.*

Though, there is certain transparency in the definition that design is a mapping of behavior to structure, it is incorrect to undermine the role of operations that achieve the mapping. Also, it appears that the earlier workers have not considered the issue of operations clearly, nevertheless, the later workers do. The distinction is clearly seen in the above definitions, as later stages, the researchers implicitly accepted the obvious role of operations in design processes. This may be, perhaps due to the finding that the operations substantially can influence the constraint

optimization and hence the final object structure. However, the design process is conveniently put as

**Design is a mapping from behavior to structure upon satisfying constraints and methodology, that explicitly called operations which is described for achieving the mapping.**

Thus, design is not only finding a suitable description of structure from its functions or behavior, in addition is a process of how this description is achieved through suitable unit operations. In other words, design is not static at description level rather dynamic as it involves operations too.

### **1.3.2 Design as an Intelligent Activity**

According to Warfield (**Warfield, 1990**) design consists of six fundamental types of intellectual activity: **Intelligence, analysis, synthesis, choice, communication and interpretation**. He maintained that these activities are critical in implementation of design. The first three activities viz intelligence, analysis and synthesis put together called conceptualization is the most essential part of understanding design and the present study aimed at investigating this phase of design activity.

As known, Artificial Intelligence is the subject that deals with understanding and emulating intelligent behavior. Intelligent behavior arises when an agent performs tasks in order to change the environment he is living in: design too demands an agent/performs tasks (against constraints) to change the previous artifact form to realize the new form of the artifact (environmental change). Thus design is said to be an intelligent behavior and hence becomes a task for AI Community.

Design is addressed in the context of system development, whether it is small or big. Design process involves cyclic generation of information. In practical design, the prospects of reaching the final object in one stroke is faint. Hence, iterative workout is necessary for approaching a solution. Generally, complex design problems are solved by dividing into smaller less complex sub-problems. However, some times design problems have a tendency of non-decomposition. According to **Bremdel & Kristainsen(1986)**, a dynamic environment, lack of resources including time, knowledge, money and designer's ignorance substantially influence the **good** design solution. They identified that design tasks are difficult besides being intelligent activities because

1. *Difficulty in formulating a specification. The ability to express the functional requirements correctly and sufficiently is weak.*
2. *There exists difficulty to structure knowledge about objects.*
3. *Knowledge is limited about functional and attributive properties of known objects.*
4. *There exists inadequate conceptual understanding of relationships between different properties of known objects.*
5. *There exists an inability to formulate, observe and interpret the most important tasks and results of an evaluation procedure.*
6. *Creativity is difficult to characterize formally.*

Engineering design, even after possessing many proven physical laws and scientific principles, is still open to individual interpretation. A same design solution

may not be guaranteed from different designers for the same design task. The way an individual expert views the task, his earlier experience and resources will greatly influence the design activity. Obviously, this shows that the design decision making is idiosyncratic and thus an art rather than a science. A further analysis reveals that design is a set of experience based activities with domain dependent facts and knowledge. By catching the experience of designers and domain knowledge and by integrating them, it is possible to implement intelligent systems on computers which aids as tools for formalizing design and for improving design practice too.

Comprehensively, design is a task that requires experienced decision making and strategies to optimize the constraints and resources. Eventhough, domain knowledge is rich in their definition, in terms of scientific principles and laws to govern, it is highly difficult or often impossible to completely eliminate heuristic knowledge. Thus, design is said to be an intelligent activity and creative/innovativeness often dominates in finding suitable solution to these tasks.

### **1.3.3 CAD AND I-CAD**

**Computers are used to automate engineering practices under the captions like Computer Aided Engineering(CAE), Computer Aided Design(CAD), Computer Aided Manufacturing(CAM) etc.** Various computer controlled machines, production robots, autonomous vehicles and comprehensive engineering databases have been developed and have shown tremendous improvement on engineering practice. On the other hand, automation of engineering decision processes particularly engineering design automation in the context of intelligent support has taken a low key.

Computer Aided Design systems available in the market supports the designer in rather very powerful way by acting as a template upon providing the necessary graphic / numerical techniques library/ algorithms or programs. Needless to say, they work at data level only i.e., they take data / numerical value / plot or graph. But to be called as any support to the designer, the support should be intelligent and should work at knowledge level. Hence, conventional CAD is not a real support to designer at all in a microsense, rather is a software consisting of various techniques to manipulate data. **SKETCHPAD(Sutherland, 1963)** was perhaps one of the earliest CAD system in current generation of CAD/CAM drafting and modeling systems.

However, during 80s considerable interest was shown by various researchers in line with AI based CAD. **I-CAD(intelligent CAD)** includes the support at knowledge level thus circumscribes a larger sphere of applicability. **Mac Callum(1990)** showed that conventional CAD progress, since then is growing steadily till 80s and there onwards became rather sluggish and readied asymptote. The cutoff point, he presented, as middle of the late 80s . Remarkably, the I-CAD which started during this time have a potential steep growth. The profile of growth as put forward by Mac Callum is as shown as Fig. 1.1 and Fig. 1.2.

In order to impart intelligence to CAD and achieve any degree of I-CAD one must understand the patterns of design tasks. Mac Callum argues that for acting as I-CAD, a system does not need to replicate human intelligence, they are required only to exhibit behavior regarded as intelligent. Necessary components of such system are knowledge-rich models of designs, the capacity for tacit knowledge and the ability to learn.

I-CAD, owing to its capability to handle knowledge of various types in ad-

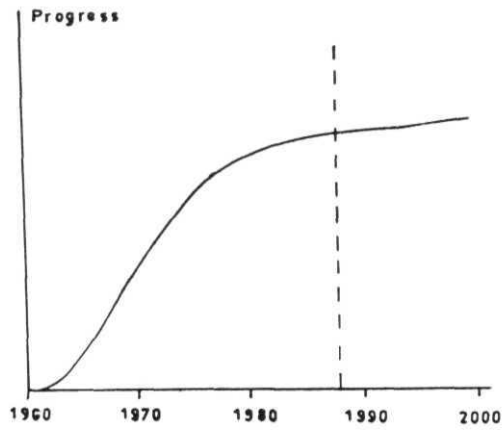


Fig.1-1 CAD Progress

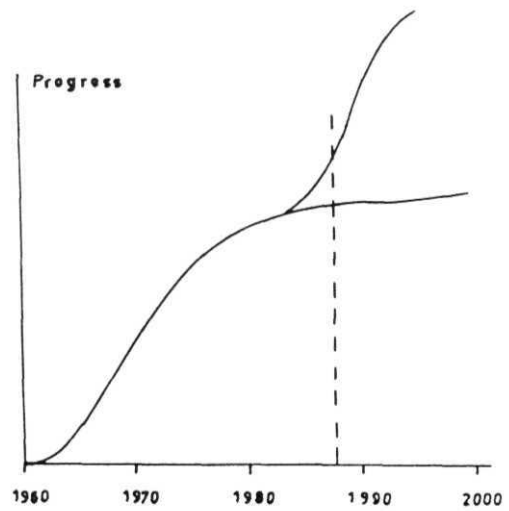


Fig.1-2 The goal of 1-CAD

dition to larger databases, graphics, geometric modeling and numerical analysis techniques has an edge over conventional CAD and thus, gained momentum from late 80's onwards. Since, design is an intellectual activity and conventional CAD has had **serious** limitations, imparting intelligence to it or in other words creating system that support design activity intelligently is more profitable in terms of both efficiency and effectiveness in design problem solving.

#### **1.4 Generic Nature of Design Tasks**

Human being **create** or designs different 'objects' or 'artifacts' chiefly to make his/her life more comfortable. Thus, he tends to change his materialistic world. The word 'object' is used throughout this study, in rather an abstract way, as a representative of an artifact or a process or a chemical formulation or an electric circuit or even fine arts and in the view of designing, no distinction was made amongst them. The realization of 'object' is through critical observation of nature and its concepts. Science is the subject which is a resultant of analysis of nature in order to uncover the laws or truths that govern the nature. On the contrary, design utilizes these laws and rules of nature to manipulate the nature and change the environment. Thus, it is imperative that design process originate cognizantly with science i.e., for every object a human being produces, there exists a need aspect and a thrust to develop his/her conceptual or scientific knowledge by replicating his learning nature.

It is easy to conceive, that the structure of the object as well as the process of series of operations to achieve, it play an important role and inevitably become the aims of the design activity. For example, consider the case of the design of a primitive flint by ancient man. The observation of the 'friction' concept behind

forest fire led to the design of the flint and a suitable process to **produce** the flint. May be, it is a **creative** design for the ancient man, nevertheless, it is 'innovative' or 'routine' for later generations, who used this flint concept for a variety of uses. Needless to say, the ancient man might have come across to know the usefulness of flame and realized the need aspect of it before he attempted to create what we are presently calling it as primitive flint. The design is routine, if designing the object is through an established knowledge. Whereas, it becomes innovative or creative if (s)he atleast apply additional knowledge to change the present 'object' structure or operations to realize the improved design object.

#### 1.4.1 Classifications of Design Tasks

Gero. (Gero, 1990) suggests two types of design tasks : **routine and non-routine**. Non-routine design tasks are further divided into creative and innovative tasks. He suggests that when all functions and structure descriptions are known before design begins, it is termed routine whereas possible design **solutions** are not known apriori in innovative kind of design. Nevertheless, in innovative kind of design, it is known that the design solution space is available. The space of possible solutions has to be extended in case of creative design. Earlier, **Brown & Chandrasekharan(1983)** had more or less same classification viz class 1, class 2 & class 3 types of design processes. For example, major inventions belong to class 1 whereas class 3 designs are akin to routine design and follows a set of well established procedures in the sense e.g., the design of an artifact is as a modification of earlier imagined object. **Warfield(1990)** upon considering design as a synthesis activity, proposes three kinds of design depending on the degree of complexity. The first kind is of negligible interaction among the components that are synthesized into a whole. Interaction which does not lead to the emergence

of new components called **non-pionic** interaction is of second kind of synthesis, whereas in the third kind of synthesis the interaction causes the generation of new components(**pio nic** interaction).

Apparently, there are broadly two opinions on the nature of design activity. One class of scientists insist that it is a classic search problem. A profound contender for this understanding is **Chandrasekharan(Chandrasekharan, 1990)**. He claims that the design problem is formally a search problem in a large space of objects that satisfy multiple constraints. What is needed is to find strategies that radically shrink search space. **Simon(1969), Maher(1990) & Boyle(1989)** also expressed similar opinion. On the other end, **Gero(1990)** argues that design is basically an exploration based activity and is the provision of form from a formless description. **Smithers et al(1990), Takeda et al(1990), Coyne(1990)** are a few who support this idea. However, **Gero(Gero, 1988)** accepted that routine design is a sort of search problem only.

To explicitly make a contrast between these two methods, the corresponding models of design at both descriptive and cognitive levels have to be compared. Cognitive models stem from the identification of human epistemological patterns while designing and these patterns will be represented as descriptive models which work at tangible level.

### **1.4.2 Design is an Exploration Based Activity**

The purpose of design is to transform function set  $F$  into a design descriptions set  $D$  in such a way that the artifact being described is capable of producing these functions.

$$\text{i.e., } F \longrightarrow D \text{ ( } \bullet \text{ is some } \mathbf{transformation} \text{ )}$$

As was said earlier, design has to **fulfill** the functions(or behavior or usefulness) of the desired object or artifact. The set of all functional needs is termed function set F. D is the set of attributes or characteristic parameters of the designed object to meet F. Thus, design is a mapping between F and D. For example, in designing a bicycle some of the functions include

1. ability to move with less friction
2. ability to take turns
3. ability to have intrinsic quality of balance
4. ability to be **driven** easily
5. ability to withstand rough roads and
6. ability to have less maintenance

The design description set contains such values as wheels, rubber tyres & tubes, handle bar, brakes, chain drive to name a few.

To model and represent the knowledge of design specifically, Gero(1990) suggested what is called design prototypes. While describing about types as a means of classifying the world, he made distinction between them as archetypes, stereotypes and prototypes. Archetypes are unique in their nature and they are first and last and there cannot be any substitutes e.g., Taj Mahal of India. Mass production of goods are stereotypes i.e., they are copies without change. Prototypes are the first upon which others are modeled. Prototype is a conceptual schema for representing a class of generalized heterogeneous grouping of elements derived from alike design cases that provides the basis for the start and continuation of design. A design prototype separates function(F), structure(S), expected **behavior**( $B_e$ )

and actual **behavior**( $B_s$ ). It also stores relational knowledge between them( $K_r$ ) as well as qualitative **knowledge**( $K_q$ ), computational **knowledge**( $K_c$ ) and contextual **knowledge**( $K_{ct}$ ).

Thus a design prototype (P) is symbolically represented as

$P = (F, B, S, D, K, C)$  where

$B = (B_s, B_e)$

$C = \text{Context}$

$K = (K_r, K_q, K_c, K_{ct}, K_p)$

$K_p = (T, P)$

P is partition

T is Typology

In this method, routine design can be viewed as design prototype instance refinement. Innovative design is a design prototype instance refinement with an adaptation of some of the knowledge concerning applicable ranges of variable values i.e., design prototype instance adoption. Creative design involves the introduction of new variables, thus is a design prototype generation. In summary, during the specific design solution, the design prototype brings together all the requisite knowledge.

Exploration based model( **Smithers et al, 1990** ) is more or less in these lines. In his view, in exploration problems, knowledge about the nature and structure of the space of all possible solution has to be obtained much prior to goals, that are to be searched, can be formulated well. What is explored essentially is the Space of Possible **Designs**(SPD).

This model starts with initial requirement  $R_i$  (Which in general/incomplete and inconsistent), reaches **final** requirement  $R_f$  (which is complete and consistent) and an associated design specification  $D$ , which is aiming at. Here, after identifying the requirements description, task is broken down (decompose), order of intermediate design goals are planned, these goals are synthesized, analyzed and optimized.

The design process,  $E_d$ , is one of exploring the SPD: a collection of concurrent and serial searches each connected by intuitive leaps, analytical assessments, synthesis, simulations, prototypes, decisions, choices, judgments etc. Acquiring knowledge about nature and structure parts of SPD and thus possible design specifications remarks in discovering the incompleteness and inconsistencies in  $R_i$ . The modifications of  $R_i$  is in a way that resolves identified incompleteness and inconsistencies and this exploration continues until a point in design space is found that fully or favorably specifies a design. This is how  $R_f$  is reached with **incompleteness** and inconsistency vanished.

The record of concurrent and serial searches called  $H_d$  (History of design process) tells what parts of design space were explored, how and why decisions were made. This  $H_d$  upon coupling with  $R_f$  and associated specifications, becomes Design description document  $D_s$ . They ruled out that design(exploration) does not just result in the coordination of some satisfactory location, as search does, it also results in knowledge about how a similar task might be done better for the next time.  $K_{dm}$  and  $K_{dn}$ , domain knowledge and design knowledge together form Design Knowledge Base and there exists certain dependency among these two i.e., in other words,  $K_{dm}$  and  $K_{dn}$  are not orthogonal kinds of knowledge (See Fig. 1.3).

An axiomatic general design theory based on axiomatic set theory was postulated to explain design process by **Takeda et al(1990)** leading to a descriptive

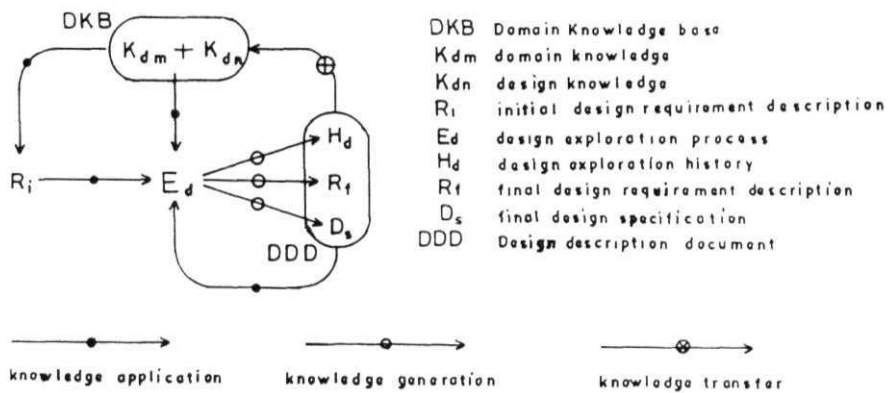


Fig. 1.3 An exploration based model of design

model which they called evolutionary design process model. Here, from the given function specification, a candidate for the design solution is selected and refined in a stepwise manner until a complete solution is obtained. The central description of the design object is regarded as a metamodel and its stepwise refinement produces the final object. Supporting this, they proposed a human cognitive model at objective and action levels. **In their design life cycle cognitive model, five successive subprocesses are identified** 1) awareness of the problem 2) suggestion of concepts **to solve** problem 3) development of design object **4) evaluation and conclusion** and 5) suggestion . In their view, first step is an abductive process, whereas evaluation is deductive. Circumscription plays an important role as to modify or redesign.

The idea of General Design Theory (GDT) is as old as in the year 1981 and was proposed by **Yoshikawa(1981)** . From there onwards, the theory **was** framed particularly to suit to intelligent CAD systems: CAD frame work guided by GDT was illustrated by Yoshikawa in 1983. An extended GDT was proposed in 1987 by Torniyama and Yoshikawa. Intelligent, Integrated and Interactive CAD(III CAD) was presented by **Akman et al** using GDT.

The basic ideas behind GDT are:

- (i) From the given functional specifications, a candidate is selected and refined stepwise until an agreeable solution is reached rather than trying to get the solution directly from the specifications.*
- (ii) Design can be regarded as an evolutionary process, which transfers the model of the design object from one state to another.*
- (iii) During the design process, new attributive description will be added and/or*

*existing ones will be modified and the metamodel is hoped to converge into the solution.*

(iv) *To evaluate the **current** state of the design object i.e. the metamodel, various kinds of models of the design object need to be derived from the metamodel in order to see whether the object satisfies the specifications or not.*

**GDT's** major achievements are mathematical formulation of the design process and a justification of knowledge representation techniques in a particular situation. Also, in the availability of ideal knowledge, a design solution is directly obtained when the specifications are described. In case of real knowledge situations, a design process can be regarded as a convergence process, if atleast some specifications appear to result in meaningful solutions. In otherwords, GDT proclaims that a rough model is a necessary condition for successful design exploration.

One of the theorems of GDT states that 'if we evolve a metamodel, we get an entity concept as the limit of evolution'. This indicates that design is a stepwise transformation process and solutions are obtained gradually refining by increasing the number of attributes and hence entity concepts. Thus, design is a stepwise, evolutionary transformation process according to GDT(See Fig. 1.4 and Fig. 1.5).

Exploration based models consider a primitive description as a basis for the start of any design process and try to refine or build upon it such that the final object is obtained with all incompleteness and inconsistency vanishing.

### **1.4.3 Search is the Process of Design**

**Chandarsekharan(1990)** viewed the task solution as three types: Problem space search(Newell, 1980 ), Well-structured algorithmic methods and ill-

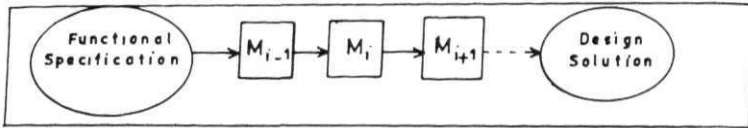


Fig.1.4 Stepwise Refinement of the Metamodel

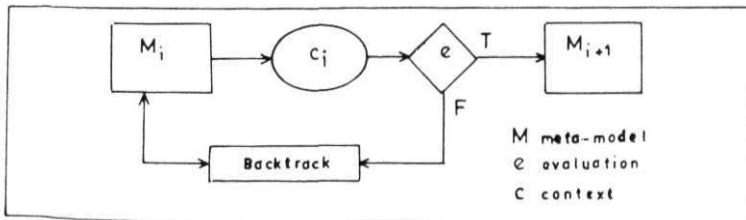


Fig. 1.5 Metamodel Evolution

structured methods. **AIR-CYL**(Brown & Chandrasekharan, 1989), **Designer Soar**(Steier, 1989) are examples that explore search spaces. *To solve* using algorithms is central idea in well-structured design solution whereas in realworld problems, the domain knowledge often sets the space of alternatives and helps control the search in these spaces.

He defined design problem by specifying (1) Functions set(F) and constraints set(C) and (2) a technology, a list of all components and their relations. He pronounced that design is domain-independent and hence, a generic activity. Design problem is formally a search problem which ~~satisfies~~ multiple constraints. The design problem solving, in his idea, is to find strategies that drastically reduce the search space and isolate the desired object. Most of the design problem solving techniques follow what is called **Propose-Critique-Modify(PCM) methods**. These methods have the subtasks of proposing partial or complete design solutions, verifying proposed solution, critiquing the proposals by identifying failures if any and modifying proposals to satisfy design goals. PCM methods identify task structure for design. It is clear that PCM methods are basically search oriented.

In PCM method, for a given goal, solution is proposed in the first step. The proposal is verified in second step and if unsuccessful the proposal undergoes critique in step 3 and in case the critique is useful in identifying the problem, the problem is modified in step 4 and returned to step 2. This is a recursive process till the goal is reached. In proposal subtask there are three methods identified: (1) Problem decomposition and solution composition (2) Retrieval of cases from memory and (3) Constraint satisfaction. Decomposition and case-based methods help reduce the size of the search spaces because the knowledge they use can be viewed as the compilation or chunking of earlier search in the design space. The third

type of above methods use a variety of quantitative and qualitative optimization techniques.

Verification, what he discussed is of two types: calculation/estimation by domain-specific algorithms(direct calculation methods) and simulation to achieve behaviors of interest and checked against requirements. Former method is not of particular interest to AI whereas the later does. AI techniques such as qualitative simulation, consolidation and functional simulation can be used for this purpose.

**Maher(1990)** , also suggested design as a search process. Three phases are identified (1) design formulation that involves identifying the requirements and specifications (2) design synthesis that involves the search for one or more design solutions consistent with requirements in design formulation and additional requirements identified during synthesis through the selection and application of operators and (3) design evaluation that involves assessing whether the goal(s) have been satisfied.

The three phases occur recursively till the expected solution is found. In this context the author's view on design is expressed as: typically a designer starts with a definition of the design problem, identifies one or more potential design descriptions and then evaluates the design. Thus, the design formulation identifies the goal(s) whereas synthesis results in describing potential design alternatives and evaluations selects the best/optimal/satisfactory solution. Variation in both the goals and the state-space descriptions as the design process proceeds in addition to the difficulty in predetermining the relevant operators are some of the issues that are not readily addressed.

Maher stated that the goals of the problem can change during the problem-solving

process, which indicate a different design space to be searched. One reason design has been difficult to implement as a search process is this change in problem definition during problem-solving process.

However, **Maher's** look out is in synthesis phase. She proposes three models for this task: decomposition, case-based reasoning and transformation. A large complex problem is decomposed into hierarchical distinct subproblems in decomposition and a model of design that directly uses design experience in the form of episodes in case-based reasoning model. In the transformation model, design knowledge is expressed as a set of transformational rules. She points out that the value in identifying models of design lies in representation of knowledge, experience and ability to choose a model that closely fits the knowledge available. Although, each model can be implemented individually, an advantage can be taken if the models are used interchangeably guided by the use of a model depending on the available knowledge.

Design as a search problem also suggests that the problem initially may not be transparent and the strategies that isolate the object behavior from other object behaviors are essential to follow one to be generated. Whether, it is PCM method or case-based reasoning the search involves the identification of primitive description upon which the goal tracking methods are adopted.

By contrasting the design problem solving techniques individually as search and exploration based models suggests the following: The design is principally a mapping from function space to structure space. However, it can be conceived that a primitive object is sufficient and necessary condition for design process to start with albeit suffers from inconsistency and complexity. This primitive object is modified with knowledge of design, knowledge of domain and knowledge

of analysis. This primitive object is decomposed into subcomponents identified by their fixed boundaries and independency to a large extent. The synthesis of the subcomponents is a process of adding one subcomponent to other by realizing their relations. At every step of synthesis, the critique and modify processes follow them mainly to compare the existing behavior to desired behavior. This process terminates when all the object functions are achieved by realized object structure.

Based on the above abstraction, it is thought that for realizing objectives of understanding and thus creating intelligent systems for design, it is required to

- (1) Study the generic nature of design as practiced in various domains*
- (2) Study the design at representative and cognitive levels*
- (3) Study whether and how computer can behave as an intelligent machine on par with the lone intelligent organism, the human being.*

The present work is, in its scope an attempt towards these lines.

## 1.5 Summary

The validity of design as a task for AI community was discussed in this chapter. It was recognized that the understanding design tasks are significant in light of their complexity, inconsistency and intuitive behaviors involved. Besides understanding what is and how design progresses in view of researchers, the present study makes a critical contrast among various models that were proposed earlier. It was contemplated here that this critique leads to constructing a global model for design activity. Both representative and epistemological levels, the critique was made, principally to have a philosophical insight on design.

The earlier work in this regard is categorized broadly in two ways: exploratory and search. Exploration based models consider the primitive description of design object as a starting point for any design to be initiated and try to refine this structure. On the other hand, isolation of desired strategy in the solution space is a principal technique in search based models. The model proposed in this study is inclined towards the former kind of models. This conclusion was arrived at **upon** identifying and defining the design task as an evolutionary synthesis process.

## **Modeling Design Process - representative Design Process Model(rDPM)**

### **2.1 Introduction**

The goal in the design activity is realization of the object. For, it requires us to develop on the available knowledge about the designing of the object. The development is from a primitive structure which seldom satisfies all the need aspects of its design. Thus, design can be called as an evolutionary process. Also, it is called synthesis process as the knowledge of primitive structure is accrued with various concepts and knowledge of components at every step of the design process to get the final structure of the object. In addition, as a science it follows established and accepted principles resulting from analysis. However, these principles are seldom sufficient to produce correct and economic design. This can be achieved in general by appropriate decisions taken by the

- human designer based on judgments and intuition. In this sense, design is an art.

Comprehensively, our understanding of the design activity may be put as a goal oriented, evolutionary synthesis process demanding judgments and decision support.

It is worth mentioning here the comments of **Mittal and Araya(1986)**

*Designing an artifact is one of the most challenging problem solving tasks performed by engineers. It is a task requiring both large amounts of domain specific knowledge as well as considerable problem solving skill.*

## 2.2 Modeling Design Task

Building an object to perform a given job upon utilizing equipment(s) and a suitable methodology of operations is termed as design activity. During this activity, designer tends to modify the environment in which he is living in. The consequences of this activity may be a better utilization of nature.

According to **Finger & Dixon(1989)**, design process models can be categorized into four varieties: Descriptive(how design is done), Cognitive(designer's model), Prescriptive(how design must be done) and Computable. **Takeda et al(1990)** proposed that a design theory for intelligent CAD is not useful when it is merely descriptive or cognitive; it must be computable. In other words, in their view a model should be able to support on-line design done in **the** area of explaining intelligence behind design task at above mentioned levels. A recent review by **Naga Prasad & Reddy(1992)** discussed understanding of **the** design as a task in the perspective of **AI** both at computable and cognitive levels. They observed that/majority of the models proposed are either coupled descriptive and prescriptive models or computable models. The former type of models are called representative models according to them. Their paper indicates that there are attempts globally to model generic design, nevertheless, there exists no complete design process model till date to explain the behavior of designers and design activity.

In this Chapter, the process of design is modeled at representative level. In order to support and give a foundation to the proposed model, a generic **Design Process Theory(DPT)** is suggested. The proposed model is called as **rDPM(representative Design Process Model)** from here onwards.

rDPM views object as a cluster of subcomponents which are linked rather diffusely with logical linkages. Identification of subcomponents and their structure, establishing linkages(relations) with other subcomponents and integrating or assembling them to achieve the final object is the subject of rDPM. rDPM evolved is based on analysis of human epistemological activity in chronological order while design action is in progress.

There are two parts in the process of design. Initially, human performer divides the design object into fragments upon imagining the final structure of the object and this chunking proceeds till undividable atomic fragments are attained. This classical divide-and-conquer policy reduces complexity of the design problem. During the process, <sup>the</sup> designer compiles lots of design knowledge including domain specific knowledge, which is essentially required to construct an object. In the course of second part of the design, the main task of the designer is to construct the subcomponent from atom components upon identifying and acquiring the knowledge that is desired to construct any subcomponent. The integration of knowledge out of these two parts is the principal methodology of the human performer. Subsequent evaluation based on engineering analysis and modification is necessary for any knowledge based tasks and so do with design tasks too. One can see from above, there are more than two phases in design though the above two parts are kernel phases of design.

There are two major problems in design: **uncertainty** and complexity. Uncertainty arises from a number of sources. Some of these may be **the** object functions, specifications and constraints. **Boyle(1989)** identified that the functions tend to be refined during the design process. He added that the initial design specifications and constraints are limited in their definitions in other words the

function set and constraint set are not completely defined before design begins. Secondly, uncertainty prevails in the original system, in the assumptions made in generating the model and in the limits of applicability of the model. Also, uncertainty exists in design techniques and in the interpretation of design data especially those in the form of graphs, plots, monograms etc. In large scale design problems, the above sources of uncertainty are unavoidable. Uncertainty can be handled by making design process iterative .

Complexity in the problem arises when there exists one-to-many or many-to-many relations between two or more parameters. Some times, the relations are sufficiently sensitive not to ignore. Problem solving expertise including design techniques and algorithms, object under design and its attributes introduce complexity in design activity. Boyle adds uncertainty to this list.

For example, he says, the confidence in a design objective's uncertainty must be stored and reasoned about in addition to the value of the designed objective itself. A classical approach to reduce complexity is hierarchical problem decomposition and recomposition. Conventionally, this is a well established system in the literature of AI.

For applying the steps of a design techniques correctly and in appropriate order, the complexity in design techniques is hindrant. In addition, some techniques use such algorithms, which have inherent complexity and need computationally intensive methods.

Besides, artifact and its attributes both may introduce complexity. For example, in composite material design, where the artifact is the composite, the nature of the formulation chemicals is sensitive to the end properties(functions). Also,

noise in the weight values sometimes influence heavily the balancing of various desired functional aspects of the material. Similarly, in the design of chemical flow reactors, the attributes of reactor such as reactor volume, flow rate, heat exchanger dimensions, etc. are mutually dependent in a complex way in addition to sensitively interrelated. The clues for attending such complexity may be available in domain knowledge itself. This complexity necessitates the number of iterations to enhance the quality of the artifact.

### 2.3 Towards representative Design Process Model(rDPM)

**Interactive Knowledge Based System(Boyle, 1989), PCM Method (Chandrasekharan, 1990), Design Prototypes(Gero, 1990), Case-based Reasoning & synthesis(Maher, 1990), Exploration Based Models(Smith et al, 1990), General Design Theory(Takeda et al,1990)** , have had different outlooks about design task, however, they have apparent common understanding about design. The following is the kernel of their learning of design and it can be confessed that it forms a basis for any further work on modeling:

**An object to be designed is expected to exhibit certain performance called functions. The aim of the design is to construct an object that fulfills the user prescribed functions. Thus, artifact oriented design is a process of mapping function set to structural characteristics set. In an ideal case, a straight solution is available which completely satisfies the mapping axioms. Nevertheless, this cannot be called a design in a stricter sense, rather it may be a search problem. The process of mapping functions to structure is always associated with a set of constraints. During the constraint satisfaction comes the question of**

**optimization and tradeoff, which are often human-specific, eventhough a weak mathematical procedure or established heuristics are available as knowledge.**

The present study, besides accepting this ideology on design, views it rather differently at the basic level itself. It argues that a further classification of function set into two subsets viz. essential function set and desired function set improves dramatically the understanding of design processes and embodies different models proposed earlier (Naga Prasad, Reddy & Kanakaraju, 1993).

Gero(1990) suggested that there are three levels of designing: creative, **innovative and routine**. When all functions and solution structures are known before design begins, it is *known* as routine design, whereas possible design solutions are not known in case it is innovative kind of design. For design to be creative, the space of possible objects has to be extended as no solution space exists. The idea of splitting up of the function set explains Gero's classification of design tasks with a higher fidelity. Making the ill-defined(innovative & creative) design tasks more explicit is an implicit objective of this work.

**Mapping function sets to structure is what the present study claims as design process model. It proposes a generic theory called Design Process Theory(DPT) for this purpose. Since, two sets of functions instead of one set are mapped to structure, it ushers to revive the understanding of the design task structure thoroughly.**

### **2.3.1 Knowledge in Design Activity**

As mentioned earlier in **chapter 1,Warfield(1988)** set forth generic design as a kind of design science. As a generic activity, design modeling considers the

common features of all specific design processes. For a successful design problem-solving or in other words, for a generic design, often it is fundamental to structure various kinds of knowledge. The organization of Knowledge Base for I-CAD necessitates the identification and assimilation of different types of knowledge, while discussing the kinds of knowledge, Warfield identified that the role of analytical methods, drawings etc. is secondary compared to the creative and conceptual kind of knowledge. Also, various researchers have tried to classify the knowledge patterns, so that, they can be better described, formalized and used.

What essentially is happening while modeling design tasks, is the organization (explicit and tacit) of knowledge and its generation and utilization. This process can be broadly called as knowledge process. It is imperative to say, that it is because of the open support of this knowledge process, AI support to the design process can be achieved to any length. **Smithers** et al (1990) evince that the role of knowledge in design tasks stems from answering such questions as :

- What *knowledge is used?*
- *How knowledge is used?*
- *Where does the knowledge come from ?*
- What *form* ~~the~~ *new knowledge fit in?*
- What *else does the knowledge get used for?*
- *How is knowledge organized and maintained?*

Definitely, there is considerable difficulty in answering the above questions in a stricter sense. Identification of what knowledge is to be acquired is forerunning

job to acquiring of knowledge it self. Quite often, identification of knowledge itself is a formidable task in design tasks.

Various workers classified the knowledge in design tasks from their outlooks. Haragu and kusiak(1987) classified knowledge into two types: declarative domain knowledge, consisting of data and facts related to domain and procedural knowledge, consisting of rules for solving the given problem. **Forbus(1988)** identifies two types of domain knowledge: Broad domain knowledge and layered domain knowledge. Past experience which he called routine design knowledge, knowledge of functional descriptions, qualitative simulation, knowledge out of procedure generation, fault tree generation and failure analysis, and knowledge of communication skills are other kinds of knowledge in his view.

**Mac Callum(1990)** discloses what is called knowledge cube. Depth of knowledge varies from facts and shallow level rules underlying commonsense knowledge. Breadth of knowledge covers variety of domains involved whereas extent of knowledge refers to the scope of design process to be included. Design activities, he suggested corresponding to a set of transformations from one part of the cube to an other. Domain knowledge, constraint knowledge, procedural knowledge, analysis algorithms and solution knowledge are some of the various kinds of knowledge involved in his view.

**Takeda et al(1990)** indicated that there are two kinds of knowledge: Ideal and Real. In the ideal knowledge, design is a simple job of mapping functions to attributes. However, they maintained that all design problems use what is called real knowledge in which design is a stepwise refinement process. Hence, to support this contention, knowledge is classified into three types:  $D_{*}$ , **design candidate description**;  $K_{oc}$  **knowledge available at the current state**;  $P_c$ , **properties**

**of the design candidate i.e. knowledge of functions and attributes.**

**Boyle(1989)** classified knowledge into three parts : imperative, declarative and innate. He explained imperative knowledge, as the one which expresses action. Formal imperative knowledge is the procedural knowledge which inturn is of two types: Explicable procedures and functions. Informal imperative knowledge takes the form of heuristics brought from experts. Declarative knowledge can be stored and retrieved but cannot be executed. Descriptions of the attributes of the objects is the formal declarative knowledge, whereas, principles based on theory, textual material such as codes, monographs etc. are informal declarative knowledge. Implicit and inarticulated knowledge is called innate knowledge. It gives man the ability to learn, exhibit commonsense and reason in unstructured ways.

**Design prototypes(Gero, 1990)** recognized that the knowledge set consists of  $K_r$ , **relational knowledge between functions, structure and behavior.**  $K_r$  provides a basis to exploit the dependencies between the variables and forms a dependency network. In addition, it contains  $K_q$ , **the qualitative knowledge**, which provides information on the effects of modifying values of structural variables on behavior and functions.  $K_{ct}$ , **the contextual knowledge**, identifies the exogenous variables for design situation and specifies that values for these variables must come from outside the design prototypes i.e. from the context.

The present study identified three kinds of knowledge (**Naga prasad & Reddy, 1992**) that are basically involved in design: **of object  $K_{obj}$ , of domain  $K_{dom}$  and of design  $K_{des}$ .** Object's functional description and attributes or specifications of it forms the knowledge of the object. Constraints on the availability of resources, qualitative and quantitative relations between functions, attributes, constraints and resources including codes, procedures, simulation generated data

and past experience asserts to be the knowledge of domain. Knowledge of design process includes the strategies to identify and organize generated knowledge, problem analysis, knowledge synthesis, recognition and establishment of relations between various parameters, linking knowledge of object and knowledge of domain, strategies to deal with uncertainty and complexity etc. Knowledge of design process is what is generic in design tasks and this is essentially what every one confesses to model.

Searching for answers to the questions raised by Smithers et al. on the role of knowledge invariably leads one to classify the knowledge as a first step towards answering them. Comprehensively, as was mentioned earlier, the knowledge involved is principally of three parts : object's knowledge  $K_{obj}$ , domain knowledge  $K_{dom}$  and knowledge of design  $K_{des}$ . It is the knowledge of design process on which much of our attention is focused due to the invariable dependency on explaining and modeling generic design tasks. Any paradigm that proposes a model for design activity, identifies and classifies design process knowledge according to it. This seems obvious in the sense that paradigm is a methodology to manipulate knowledge and undeniably any paradigm should acknowledge the types of knowledge(s) it has to handle at every step. The present study, while suggesting a design paradigm, later treats the design process knowledge.

Finally, knowledge from scientific, technical and sociological developments said to influence the function specifications, design process alternatives etc. And this additional knowledge which aids in designing of especially innovative and creative objects is to be acquired on a continuous time frame.

### 2.3.2 Constraints in Design Activity

**Sriram & Maher(1986)** claimed that the functional specifications, limitations of the resources and the design criteria can all be treated as constraints on the design of/object. Hence, constraint satisfaction and their integration from various sources is the process involved in design tasks according to them.

**Sriram & Maher(1986)** attempted to define constraint as a relationship between variables. They tried to give a formal description of nature of constraints. Five types of constraint are identified by them: synthesis, interaction, *causal*, parametric and evolution constraints. Synthesis constraints effect the generation of feasible configurations whereas interaction constraints arise due to the interaction among subobjects. Feasibility of alternatives, response of subsystem to **the** changes in environment are cases of *causal* constraints. Parametric constraints, as the name suggests occur during the parameter selection and process evaluation constraints originate in ranking the alternatives.

In routine design, **Brown & Breau(1986)**, identified various kinds of constraints involved and argued that they differ not by their content but by their location, method of activation and purpose. Implicit constraints, **inplace** constraints, inherited constraints and accumulation constraints are constraint types categorized by them. Implicit constraints are built-in the design knowledge and are satisfied automatically without being explicitly expressed. Inplace constraints act to test the relationships among attributes and are explicit. They suggested that the general purpose of inplace constraints is to test the progress of the design. These are characterized by their location. Inherited constraints arise after a design decision has been made by the designer and correspond to inheritance of the constraint from the generic knowledge, triggered by adding a value to the

instance. An accumulation constraint monitors a value which has several pieces of design knowledge contributing to it.

Constraints play an important role in the design process. Identification and classification of constraints helps in planning the strategies to satisfy them. They demand sufficient domain knowledge to satisfy and strategies to simplify, optimize or trade off. Recognizing the constraint(s) is reckoned with as a major leap towards problem solving, since, often it is considered as a bottleneck. Constraints arise at every design step, whether, it is problem formulation or evaluation and need to be satisfied.

### **2.3.3 The Role of Operations**

Though design activity is said to be a mapping from functions to structural attributes, often the path in which the functions are achieved from initial model decides the final object structure. This is more pronounced when fulfillment in toto of function set is a distinct chance and design leading to optimal / acceptable object structure is always the outcome. Operations play a critical role when design alternatives are too many but none are absolute solutions, e.g. take the case of chemical synthesis. The procedure adopted considerably influences the product chemicals formed, their type and amount. The acceptable levels of a function decide the operations that have to be suggested to achieve the agreeable final object(or principal product).

## **2.4 Theory of Design Process**

Conventional CAD supports human designer by providing a template for simulating and modeling design the **objects**. **In** this case, the role of the human de-

signer is quite large. CAD merely helps him in manipulating vast data varied both in length and breadth i.e., CAD operates at data manipulative stratum. However, it is not adequate for solving design problems, which is basically an intelligent activity. Also, since design is a decision intensive task, conventional CAD should be augmented with intelligent interface for it to be called as intelligent CAD(I-CAD) system. I-CAD processes knowledge in addition to data and includes conventional CAD as a subroutine Thus it is more effective than CAD alone. Like any other intelligent task, design task too demands knowledge to be expressed explicitly and any system that supplies I-CAD environment should retain this point in view. Hence, it is required to have a theory that explains knowledge structures and problem solving strategies suitable for design tasks, thus, visualizing the potential of I-CAD to any degree. In the present study, a theory is proposed to explain design activities in real knowledge.

#### **2.4.1 Design Process Theory(DPT)**

Generally, design process is described at two planes: one at the knowledge plane and another at the parametric plane. At the knowledge plane: object knowledge, domain knowledge and knowledge about design methods are processed. At parametric plane, the parameters that represent quantitatively the domain knowledge are manipulated. In other words, the latter is driven by quantitative **principles** and equations, whereas, the former operates on qualitative physics. Evidently, we think qualitatively in knowledge plane but practice design quantitatively i.e. in parametric plane. Both methods are used simultaneously and it does infer that both planes are interspersing and it is not possible to consider any one plane in isolation for design.

Complete parametric design is a case of routine design as the goal state and solution state are well defined. The remaining job is to simply substitute the relevant parameters in relevant mathematical identities and solve them to fix the specifications. As stated earlier, the real world problems seldom calls for this type of design.

The interdependence nature of parametric and knowledge levels of design activity can be used effectively in DPT in order to express the theory more explicitly. Each term in **knowledge** level description is related to a corresponding term in **parametric** level description(Parametric plane). **Table 2.1** illustrates some of the vocabulary generated in this respect. When one level of description is taken up, it automatically infers about the corresponding other description.

In general, the task of the designer is to design an object to satisfy a set of objectives, which is called a set of functions. As mentioned earlier, two types of functions are found: Principal and Secondary. Principal functions are some sort of essential functions and the set of all principal/essential functions is denoted by  $F^e$ . Similarly,  $F^d$  is a set of all secondary functions which also is called the set of desirable functions. Needless to say,  $F^e$  and  $F^d$  are subsets of function(F) set. This classification is necessary as the innovative and/or creative design starts with satisfying all  $F^e$  elements and includes as many elements of  $F^d$  as possible.

For example, take the case of designing **hairpin heat exchanger(Purohit, 1983)(see Fig. 2.1)**. It is extensively used in a variety of services particularly when one stream is gas or viscous liquid or when flow rates are small. (Heat exchanger is an equipment that exchanges heat from hot fluid to cold fluid). Also, **these exchangers** are preferred due to their low cost in high pressure services. The essential functions of the object under design(heat exchanger) is to heat/cool the

Knowledge of qualitative representation	Knowledge of quantitative representation
<p>1. Functions &lt; <i>purposes</i> &gt;  <math>F=(f_1, f_2, \dots, f_i, \dots)</math>  <math>f_i</math> is <math>i^{th}</math> element in the set of all functions</p> <p>2. Structure &lt; <i>shape</i> &gt;  <math>S=(S_1, S_2, \dots, S_i, \dots)</math>  <math>S_i</math> is <math>i^{th}</math> element of the set of structure components</p> <p>3. process technology  &lt; operations to perform &gt;  on the components  <math>O=O(O_1, O_2, \dots, O_i, \dots)</math>  <math>O_i</math> is <math>i^{th}</math> element of the set of all operations</p> <p>4. constraints &lt; on structure &gt;  and constraints &lt; <i>on operations</i> &gt;  <math>CS=CS(CS_1, CS_2, \dots, CS_i, \dots)</math>  <math>CO=CO(CO_1, CO_2, \dots, CO_i, \dots)</math>  <math>CS_i, CO_i</math> are <math>i^{th}</math> elements of structure constraints and operation constraints sets respectively</p>	<p>parametrics of functions  <math>f_i=f_i(f_{i1}, f_{i2}, \dots, f_{ij}, \dots)</math>  <math>f_{ij}</math> is <math>j^{th}</math> parameter that defines <math>i^{th}</math> function <math>f_i</math></p> <p>parametrics of structure  <math>S_i=S_i(S_{i1}, S_{i2}, \dots, S_{ik}, \dots)</math>  <math>S_{ik}</math> is <math>k^{th}</math> parameter that defines <math>i^{th}</math> structure <math>S_i</math></p> <p>parametrics of operations  <math>O=O_i(O_{i1}, O_{i2}, \dots, O_{il}, \dots)</math>  <math>O_{il}</math> is <math>l^{th}</math> parameter that defines <math>i^{th}</math> operation</p> <p>parameters of constraints  <math>CS_i=CS_i(CS_{i1}, CS_{i2}, \dots, CS_{im}, \dots)</math>  <math>CO_i=CO_i(CO_{i1}, CO_{i2}, \dots, CO_{in}, \dots)</math>  <math>CS_{im}, CS_{in}</math> are <math>m</math> and <math>n^{th}</math> parameters that define <math>CS_i</math> and <math>CO_i</math> respectively</p>

Tablet 1 Correspondence of knowledge of qualitative and quantitative descriptions.

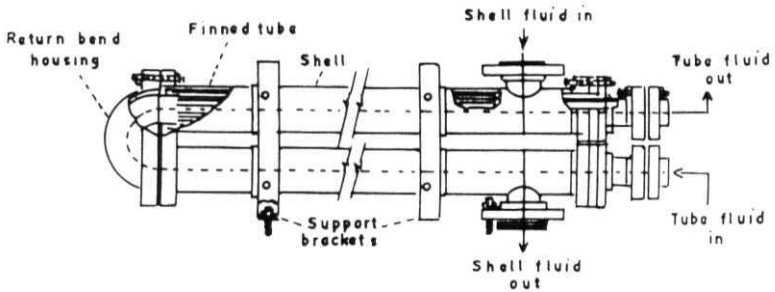


Fig 2 1 Hairpin ( doublepipe ) heat exchangers

given process stream(usually, the fluid which is of relatively more importance is called process stream and the other is called heating/cooling fluid) with low pressure drop, with ease of maintenance compared to other types of heat exchangers. The desired functions are to have lower cost, more efficiency etc. The idea of finned tubes obviously reduces the cost and hence is an innovation. The functions of hairpin exchangers is set  $F^e$ (essential functions) and functions for finned exchangers is  $F^e + F^d$  as it is a development from hairpin exchanger. Satisfying  $F^e$  completely is the case of routine design and designer's job is simply to bring relevant domain knowledge depending on the context.

**Proposition 1:** Design is a *process* of satisfying  $F^e$  completely and to an extent  $F^d$  depends on the ability and context.

In case of a tradeoff between the competitive options the elements of  $F^d$  can be ignored to that of  $F^e$  elements. However, the efficacy of the design lies in satisfying as many elements of  $F^d$  as possible.

For example, describing merely the essential functions of hairpin heat exchanger is of little use, rather it is required to **prescribe** the values atleast vaguely. In this case the temperature reduction required , allowable cost(elements of  $F^e$ ),corrosion level etc. have to be specified.

In order to achieve the goal, the functions set should be transformed into parametric set. There exists **one-to-many** relation between these two sets, i.e. there exists one or more parameters that define one function and there may be common parameters between two functions. The parametric set of functions possesses two subsets  $f^e$  and  $f^d$  respectively corresponding to qualitative set of functions  $F^e$  and  $F^d$ .

$$f = f^e \cup f^d$$

and

$$F = F^e \cup F^d$$

$$F^e = (F_1^e, F_2^e, F_3^e, \dots, F_m^e)$$

$$f^e = (f_1^e, f_2^e, f_3^e, \dots, f_n^e) \quad m \leq n$$

similarly,  $F^d$  &  $f^d$

**Proposition 2** : *An element of  $F^e$  can be defined by a number of parameters which also may inturn define fully or partially an element of  $F^d$ .*

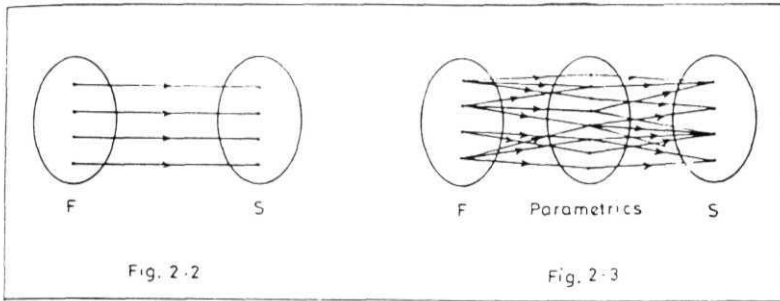
**Lemma 1 :** *There may be common parameters existing between two elements of  $F^e$ , between two elements of  $F^d$  and between elements of  $F^e$  and  $F^d$  that simultaneously define the elements either partially or fully.*

**Lemma 2 :** *Other terms that are shown in **Table 2.1** viz structure, operations and constraints follow the same course of propositions 1 and 2 including Lemma 1.*

Proposition 2 including Lemmas 1 and 2 outlines the relations between  $F^e$  and  $F^d$ . An important consequence of this proposition is the identification of common parameters which initiates the human's innovative or creative activities to develop on the system. In the heat exchanger mentioned above, for instance, the parameter, the number of pipes required for a given functional requirement is common to both  $F^e$  and  $F^d$  and it provoked the thought of reducing number of pipes even by keeping total HT(Heat Transfer) area constant by using fins on each pipe(increases HT area per each pipe).

In addition, the outcome of the identification of parametric description of design as a separate plane to that of knowledge description, is the view on design process itself. Much of the AI related work on design expresses design as essentially the process of mapping functions' set to structure sets. Fig. 2.2 and Fig. 2.3 contrast this referring the parametric relations.

The complexity that prevails in design implicitly is large due to one-to-many mappings with parametrics even though apparently the elements of F and S are mapped one-to-one.



#### 2.4.1.1 The Origins of $F^c$ and $F^d$

The object to be designed can be viewed as a logical combination of components. In artifact or equipment design, the decomposition of artifact into components is easy to conceive as the decomposition can be imagined in physical terms. However, in a few design processes decomposition can not be possible physically, and in these cases the principal job can be decomposed into component jobs i.e. into logical fragments. Each component is further divided into subcomponents and so on and the division continues until we get undividable fundamental fragments called atom components.

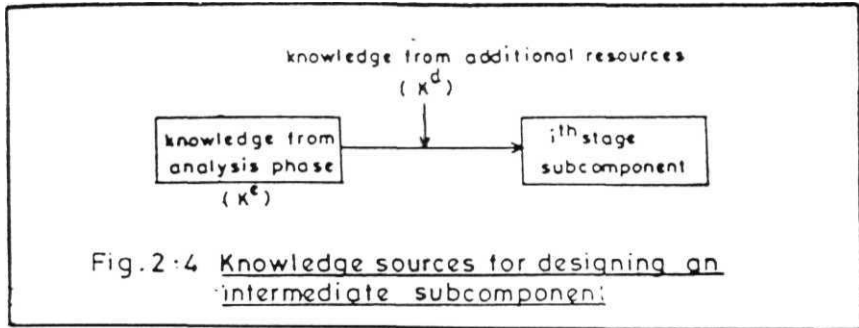
Human designers by virtue of their genesis possess the ability to visualize the primitive form of the object and ways of decomposing this primitive structure into multitude of parts presuming the structure as a collection of structural subcom-

ponents or as a collection of logical subcomponents depending upon the context. It is difficult to exhibit by a machine this type of human visualization which originates mainly through intuition. Nevertheless, knowledge of functional aspects of the object including the structure, operations and their constraints are generated during this step. This is precisely what is called essential aspects of the design (functions, **structure, operations and constraints**).

During the analysis of primitive form of the design of hairpin heat exchanger, much knowledge is acquired about how and where outer pipe is physically connected to inlet system. The problems involved in designing the outerpipe-inlet system i.e. the constraints, structure of the outerpipe-inlet system are understood at least qualitatively. Obviously, this qualitative knowledge is fundamental for design synthesis.

While satisfying as many constraints as possible, the designer opts or chooses to satisfy particular **constraint(s)** over other constraint(s) when there is a question of competition or lack of domain knowledge. Elements of set  $F^d$  stems from the designer's thrust to overcome as many constraints as possible which were considered but ignored earlier.  $F^d$  is fulfilled in general from inductive knowledge or from other knowledge resources.

In the case, where cost is a severe constraint in hairpin heat exchanger design, the designer is forced to see the alternatives to reduce the number of pipes and length and diameter of each pipe. As a result, he arrived at fins for each inner pipe arrangement which increases the heat transfer area per each tube or in other words for a given HT area, the number of tubes reduces.



**Proposition 3 :** *The knowledge generated out of the analysis of primitive form of the object becomes essential knowledge for designing.*

A subcomponent is formed making use of two sources of knowledge(Fig. 2.4): one is the knowledge from analysis of primitive structure and the other is the additional knowledge which the designer wishes to augment, i.e. the design may be considered as an art of arriving at an artifact of given specifications/variables from a primitive form of the object with stated specifications and is a process of improvement/modification.

The knowledge that the hairpin exchanger consists of various subcomponents and are linked in a specific way to fulfill certain functions is the knowledge generated from the analysis of primitive form of it. This knowledge may be inadequate for complete design, however is essential.

**Proposition 4** : *The initial model of the object is a necessary and sufficient requirement for complete design.*

As design is basically a modification process, the condition of necessary requirement is obvious.

The case of sufficient requirement arose, since, if the artifact's primitive structure is known, the complete structure is obviously made. Though, it can be argued that there may be atleast one constraint that can not be satisfied with available knowledge and hence, the sufficiency condition for design should be on **the** knowledge of constraint satisfaction but not on the initial model, it lacks gravity as explained below.

<sup>(A)</sup>  
In majority of cases, a complete satisfaction of constraints is not possible in a stricter sense due to lack of knowledge. However, a higher margin of allowance is considered accounts for the effect of that constraint, this allowance can be prescribed by suitable judgment or heuristics based on experience. Hence, constraint satisfaction cannot be a sufficiency condition. Rather, the sufficient condition is an initial model. If initial model(primitive model) is available then one can take decisions or judgments on designing.

Proposition 4 emphasizes the significance of initial model. It can be said that the entire design process depends on the availability of initial model and knowledge generated from its analysis. Nevertheless, this primitive model is a guide and refinement over it arrives at the final stage.

**DOMINIC(Howe et al, 1986)** which was developed for civil engineering structural design, also suggests the necessity of rough initial design(primitive model in our argument). According to (authors, DOMINIC either computes a

rough initial design from routines supplied by the domain expert or accepts a design provided by the user who specified the problem . Similarly **AIR-CYL, an expert system for designing Air Cylinders(Brown & Chandrasekharan., 1983; 1986)** obviates the need for primitive model. **Gero et al(1990)**, while proposing **design prototypes** suggested for initial model.

**Proposition 5** : *For every design task, there exists a set of atomic level sub-components which sure integrated to form next higher sub-components and so on. The subcomponents are logical segments and are characterized by their independence and fixed boundaries.*

This proposition forms a basis for knowledge representation as a task tree. Sometimes, however, the same subcomponent may be necessary for designing two different subcomponents that result in two different paths. In such occasions, the design of the subcomponent needs to be duplicated instead of designing once again.

An **atom** component is a well defined part of the object having its own boundaries(physical/logical) or attributes and has specific functions to fulfill. A boundary is in such a way prevails that the neighboring atom component is linked to atleast one other atom component. The link is again either physical or logical.

In the earlier mentioned example of hairpin heat exchanger, its physical form is decomposed into outer pipe, inner pipe(s), shell, supports. Outer pipe is further fragmented into outer pipe, inlet and outlet connections. Assuming, these three are atom components, it is not difficult to visualize the boundaries of each of them. It is clear that the outer pipe is physically linked to atleast one atom component i.e., to either inlet or outlet system. For any parent-subcomponent, if there exists

only one child subcomponent, the boundary is in such a way occurs that it can be linked to any neighboring subcomponent or its atom<sub>i</sub> component. That is, the primitive structure decomposition results in a sort of component-tree with 'and' linkages.

**Proposition 6 :** *Both the specifications of the atoms and the logical connections with other atom(s) influence the quality of the parent component.*

This proposition is useful for deciding the adequacy of the designed subcomponents as the quality of a component is related to subcomponents' specifications and their interrelations. Evaluation requires knowledge of purpose(functions) and knowledge of structure and this proposition highlights this point.

If there exists any uncertainty in the definition of atom(s) or in their interrelations, this proposition suggests that it reflects in the design of parent subcomponent. The additional knowledge, while synthesizing the design object eliminates the uncertainty.

#### **2.4.1.2 Criteria for Object Decomposition**

If there exists knowledge about how to decompose the primitive object as in case of routine design, the designing problem becomes highly simplified. In case, if it is not existing, parameters of the essential functions may be taken as a guide and the one that principally satisfies at least one element of  $F^e$  be designated as a component.

Designing an atom is purely through domain design knowledge. Thus in any

design process, the designer should have either the designed atom or sufficient knowledge to design the atom.

One, two or many atoms combine together to produce an intermediate form. Recall that **during** the analysis part of domain, it is said that the parent node of the atom component (parent subcomponent) consists of a group of atoms. But, however, while integrating, it may or may not yield the identical parent node, in the sense that, the description of a particular subcomponent while analysis of the primitive model of the object and while synthesis of the same considerably differs and the difference is two fold. The analysis produces information in knowledge plane and essentially  $F^e$ , whereas, in synthesis phase, the work out is in parametric plane in routine design, both parametric and knowledge planes in innovative and creative designs and includes extra knowledge of  $F^d$  in addition to  $F^e$ .

In hairpin heat exchanger, the analysis of inner pipes results in pipe perhaps lengthily. However, while designing, not only the geometry of the pipes is specified but also specified the thickness of pipe, how it is connected to other parts and how it is supported.

To be more emphatic, let us consider one more example, the case of designing a continuous mixer-reactor for liquid-in-liquid (**Levenspiel, 1972**); (**Harubs et al, 1985**); (**Nauman et al, 1983**). For the same purpose, a batch mixer can be used, wherein, all materials are added in a vessel, stirred and withdrawn. This mixing method is improved by making the process continuous so that the productivity increases. For brevity, it is not rigorously working for a liquid-liquid mixer but only outlining the phenomenon.

The initial imagination of a continuous stirrer is as in Fig. 2.5. This can be

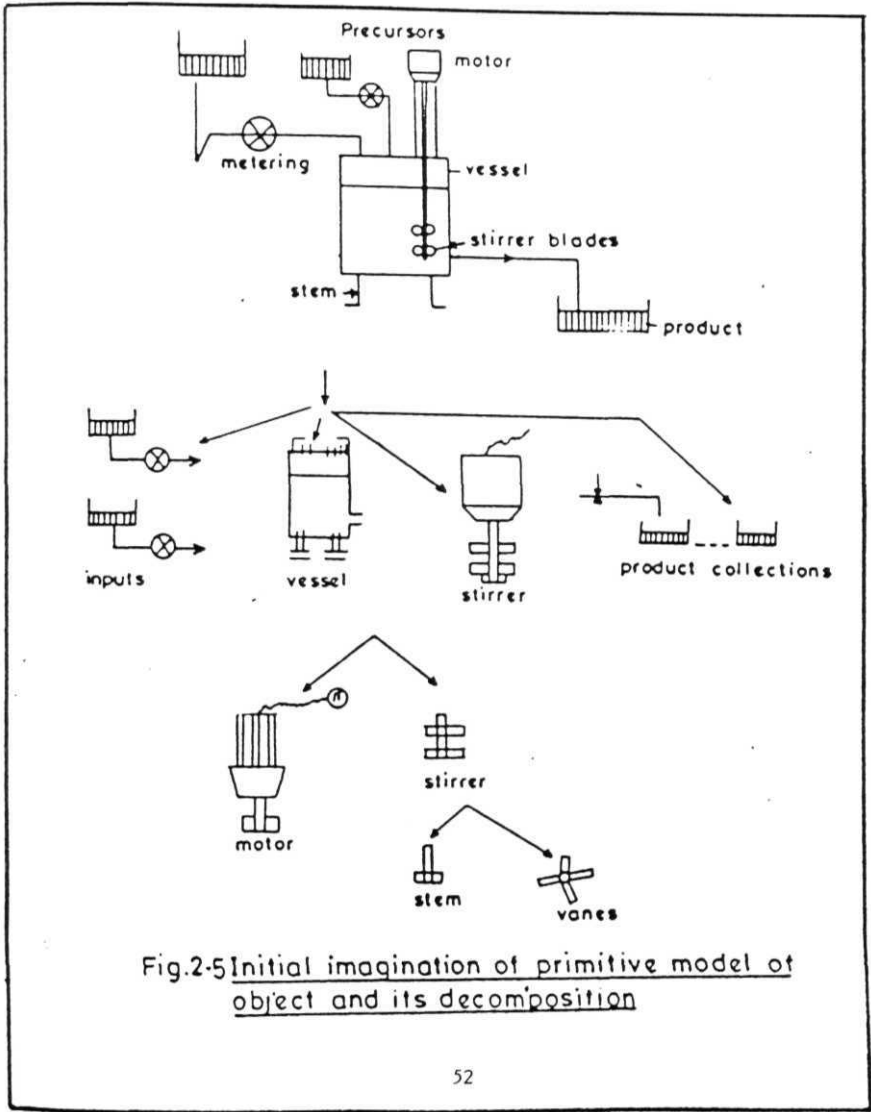


Fig.2-5 Initial imagination of primitive model of object and its decomposition

through experience or induction from other subjects. Chemical reaction engineering traditionally adopts this stirrers for liquid-liquid mixing. A logical decomposition of the object reveals that

It is composed of

1. a continuous input of precursors
2. a mixing vessel
3. stirrer mechanism
4. continuous output of product

Each of these subcomponents is related apparently to others but can be identified as independent in their functional contribution to overall object. Also, each of the subcomponent is further divided into smaller components e.g., the stirrer subcomponent is divided into the stirrer part and the motor part. The stirrer subcomponent is further decomposed into stem, blades, and blades are further decomposed into vanes(Fig. 2.5).

The decomposition mentioned above may be aided by general knowledge or qualitative physics. For example, liquids will be agitated by a **stirrer**, a power source is necessary to rotate stirrer etc. In case this knowledge is not available, one has to select a subcomponent in such a way, that it satisfies at least one functional aspect of the overall object.

For the sake of **argument**, consider a branch of Fig. 2.5, the stirrer. The elements of  $F^e$  as an outcome of analysis are the decomposed parts, functions of each part, description of each part, how each part is connected to the other, dimensions

and engineering specifications of each part. Due to the localized concentration of exothermic energy released by reaction leads to hot spots. The hot spots are not preferred, since, the reaction is uncontrollable. Localization of energy may be due to poor heat transfer characteristics of the liquid mixture. Let this constraint be tackled at analysis part by increasing mix speed and by reducing mix time. However, they may not fully contribute to overcome the constraint of hot spots as more mix speed imparts significant mix energy which in turn increases the reaction. But, the decision of increase in rpm to certain extent works. Now, during the synthesis part, the designer checks whether any improvements are possible, including other than increase in rpm, which tackles the **constraint** more effectively.

Knowledge of cold shots (adding part of cold raw material in the middle of mixed or any other geometric location) or recycling part of the cooled product etc., may be considered for overcoming the constraint. This knowledge **becomes knowledge** of the desired aspects. This may improve/change the vessel subcomponent **design**. Or some times, many subcomponents such as stirrer or input lines or output lines may be changed. In addition, the subcomponents such as external cooler **etc.** may be required. Obviously, it can be shown that the subcomponent description in analysis phase substantially varied compared to that of synthesis phase and variation as mentioned previously lies in the satisfaction of  $F^e$  and in the extent of satisfying  $F^d$ .

rDPM advocates that the actual design process starts at the synthesis phase. A preliminary analysis of design objective such as an artifact or a chemical process provides a template for synthesis to begin. Synthesis of subcomponents as directed by analysis description initiates with a process of finding logical interlinks between two subcomponents at a time and operations to interlink. Thus, the integration of

elements involves two steps: (i) establishing logical relation between two or more nodes at knowledge plane as well as parametric plane and (ii) actions/operations to establish that relation.

**Proposition 7** : *There exists between two elements a logical relation such that the result leads to its parent node and if no relation prevails, the elements become non-interactive.*

Only those combinations leading to a parent component are said to be productive, whereas, other combinations are said to be in vain. Obviously, the inlet system and outerpipes in hairpin heat exchanger are related physically and **their** relation leads to outer shell system. Nevertheless, there exists no relation between outlet system and inlet system, hence their combination is not productive even though they both are elements to the same parent subcomponent.

Implicitly, this proposition suggests that there is certain hierarchy prevails in the arrangement of elements to be designed. The knowledge required to keep them in hierarchical order is generated while analysis of primitive structure of the object under design. In hairpin heat exchanger the arrangement order is to design outerpipes initially, later the inlet system and finally the outlet system. Even in continuous mixer example, the vanes have to be designed earlier to stem. **EDISON, expert system for mechanical devices design, (Dyer, Flowers & Hodges, 1986), DOMINIC(Howe et al, 1986), SMECI and expert system for civil engineering design with special references to Harbour design(Neven & Haren, 1986)** suggests for hierarchical arrangement of elements of subcomponents.

**Proposition 8 :** *For every productive logical relation there exists a logical action/operation to achieve/establish it.*

Unless some action is done, no two elements though interactive cannot logically lead to a successful parent subcomponent. This unit operation is an essential criteria for deciding the best/optimum design method. In synthesizing a component from its child node subcomponents, there may exist a number of relations between subcomponents and for each relation there may be a number of operations. The selection of a suitable relation and operation is a process of selecting an **element** in relations-operations matrix using essentially domain **knowledge**. The number of child nodes, though, arrive chiefly during the process of primary analysis, however, can be enhanced by means of induced knowledge from allied fields or by other knowledge sources. This inductive knowledge results in innovative and/or creative design. The extra children nodes provide a basis for realization of as many elements of  $F^d$  as possible.

DPT recognizes the significance of operations involved in achieving the subcomponent(s) as much as selecting the function-attribute formulations. Many constraints demand the necessity of proceeding design in the easiest or in a way low in cost i.e. selecting an optimum/easier operation.

The proposed theory of generic design processes suggest that design is basically an evolutionary activity. It evolves from a hypothetical primitive model which acts as a template upon which the object is built. In case of routine design the primitive model possesses many/almost all features of the final object(i.e. ultimate goal). Hence, for routine designs, the primitive model is conceived through earlier experience on similar objects or by analogy with similar object's design. Atleast in

this context, the expert systems like **AIR-CYL(Brown & Chandrasekharan, 1983, 1986)** or **DOMINIC(Howe et al, 1986)** assume the availability of primitive model(initial rough design in according to them) as granted arc routine designs.

On the other extreme, creative design starts with human intuition and knowledge about even the primitive model is scanty and feeble. Often analogy with nature(e.g.creating aeroplane analogical to birds flying, creating cybernetics analogical to human's or other beings etc. or extraordinary imagination of utilization of physics principles(reaction forces leading to rocketry, friction leading to flints, electromagnetism leading to telephone etc.) initiates the creative design process. As this kind of making even the primitive model is intuitive and imaginative, creative processes become themselves idiosyncratic and unexplainable and hence, cannot be automated with available techniques.

Innovative design is in between these two extremes. Here, the extra knowledge is added at each node of the subtask tree. Acquisition or identification of this knowledge is the crux of innovative design. Quest for developing the existing subcomponent(s) is the foundation for innovative design. In this case analogy or previous expertise or use of first principles are the principal methods of how to develop on the existing system i.e. knowledge essential for innovation in design. Recognition of locations where newer methods/developments/new ideas can be adopted is a first and basic step towards understanding and modeling innovative design tasks.

Design Process Theory is proposed in order to model the design process. DPT is significant in understanding generic design activity more explicitly. It recognizes the locations of where and what knowledge is needed, generated or processed. In

other words, knowledge processing is the key issue which was explained using DPT.

## **2.5 representative Design Process Model(rDPM)**

Design Process Theory identifies the function set as a composite of two subsets: the essential functions and the desired functions. It suggests that functions, structure, operations and constraints to achieve the structure should be dealt with both in knowledge plane and parametric plane descriptions. Identification of disjunctiveness aids in the realization of essential and desired aspects of knowledge. In addition, it forms a basis for criteria of decomposing **the** initial **primitive** model. Design Process Theory is postulated chiefly to visualize the design **description**. A representative model called(rDPM) based on this theory is proposed to deduce the final object structure from initial primitive object structure. It is recalled that the primitive structure though lacks many features as that of the final object, is a necessary and sufficient condition for arriving at the final goal.

### **2.5.1 Description of rDPM**

In rDPM, it is to be proposed that design is a gradual **transformation** from function space to structure space. This model is conceived to be **useful** for building a framework that can be implemented as an intelligent assistance. Knowledge about how to perform the design process and tools applicable including actions to be carried out is provided to the designer at each design stage. **Principally** rDPM is used to guide these knowledge processing and inference mechanisms.

The basic ideas behind rDPM are derived from recognizing design as an evolutionary synthesis process. Initially, the primitive model is selected in the way explained earlier. This is called **select phase**. The process of analyzing the

primitive model results in object decomposition i.e. into component-tree with a hierarchy and with 'and' linkages. As previously argued, the byproduct of decomposing into tree structure is the (Fig. 2.5) reduction of complexity that prevails in design tasks. Much of the knowledge especially knowledge of how to design, what to design and how the relations exist between subcomponents at various levels of design are generated. This aids in planning and synthesis of components. rDPM views this decompose **phase** as a crucial step in generic design tasks.

Once the subcomponent is to be designed is identified, the task is to design it i.e. find attributes which fulfill the functions. This is an **integrate phase**. Here, knowledge from analysis( $K^e$ ) is integrated to knowledge from additional resources( $K^d$ ) to produce the desired need (Fig. 2.4). Atom component as was defined earlier is the smallest component possible which exists before design problem begins and cannot be further fragmented..

It is recollected always that the integrate phase works out both at knowledge plane and parametric plane interspersingly. It is showed later in **chapter 4**, which describing a prototype expert system in electrical power supplies domain, how the integration leading to subcomponent is practically performed simultaneously at knowledge and parametric planes. As told in DPT only those components which are productive (propositions 7 and 8) are integrated to produce parent subcomponent. Also, in integration phase, the seeds of innovation are exploited and with the aid of user, the innovative design can be activated and can be formalized.

The integrated subcomponent should be able to fulfill all the functions which are specified. However, seldom it will do so, because of the uncertainty and complexity exist in every design process step. But the completeness of design of any  $i^{th}$  subcomponent is controlled heavily by the degree of functional satisfaction to the

user needs. In other words, in every designing of subcomponent, we tend to get an optimum/satisfied structure rather than perceptual subcomponent. This constraint satisfaction for maximum fulfilment of functions set of  $i^{th}$  subcomponent is judged in scrutiny step.

Once, the subcomponent is synthesized in the above way, it is tested against its performance of satisfying at least  $F_c$  and this test procedures include engineering analysis both at knowledge and parametric planes. The course of designing subcomponent follows much the same way as complete design of object. The criteria for testing is the doctrine that every component possesses a few functional needs that are to be essentially met and component **parameters** should attain corresponding expected specification(s).

**scrutiny phase** involves the engineering analysis and knowledge for evaluation. Often, analysis has to be carried out because the interests are always centered at final object's structure and functional aspects of it, rather than the subcomponent under current study, i.e. it is not required to achieve an ideal subcomponent rather require a best final object. In evaluation process, always the effect of subcomponent design is prognosed. In case, the design fails in scrutiny phase, the subcomponent is either modified or redesigned i.e. altered.

**alter phase** is a feedback recursive process which is performed till the final goal(either subcomponent design or final object design) is reached. If the subcomponent could not meet all the essential specifications, if not desired **specifications**, the subcomponent design undergoes this process of modification.

In alter phase the reason for its failure is thoroughly analyzed using domain knowledge and new relations and/or operations are selected. This process runs

recursively till the subcomponent meets the required specifications.

A parent subcomponent is built using these subcomponents and exactly, the same procedure is adopted in evaluating the parent subcomponent. However, there is a basic difference. The scrutiny and alter procedure for parent subcomponent results in the development of its child node but sometimes need thorough look back even to the leaf nodes/elements of artifact. At all levels of design synthesis, modifications may be needed and the case of rigorous shake up of **the** entire synthesized subcomponent cannot be ruled out.

Fig. 2.6 shows comprehensively the representative Design Process Model. Due to the scrutiny and alter phases, the design process becomes a recursive activity. Since, it is found that parameters that satisfy the function set more optimally, the design process is said to be explorative. Integrate and alter phases perform such activities as explores those parameters that enhances design object satisfaction level. In summary, for generic design tasks, it is proposed that **the** paradigm **select-decompose-integrate-scrutiny-alter** is suitable. This is precisely the objective of the rDPM (Naga Prasad, Reddy & Kanakaraju, 1993).

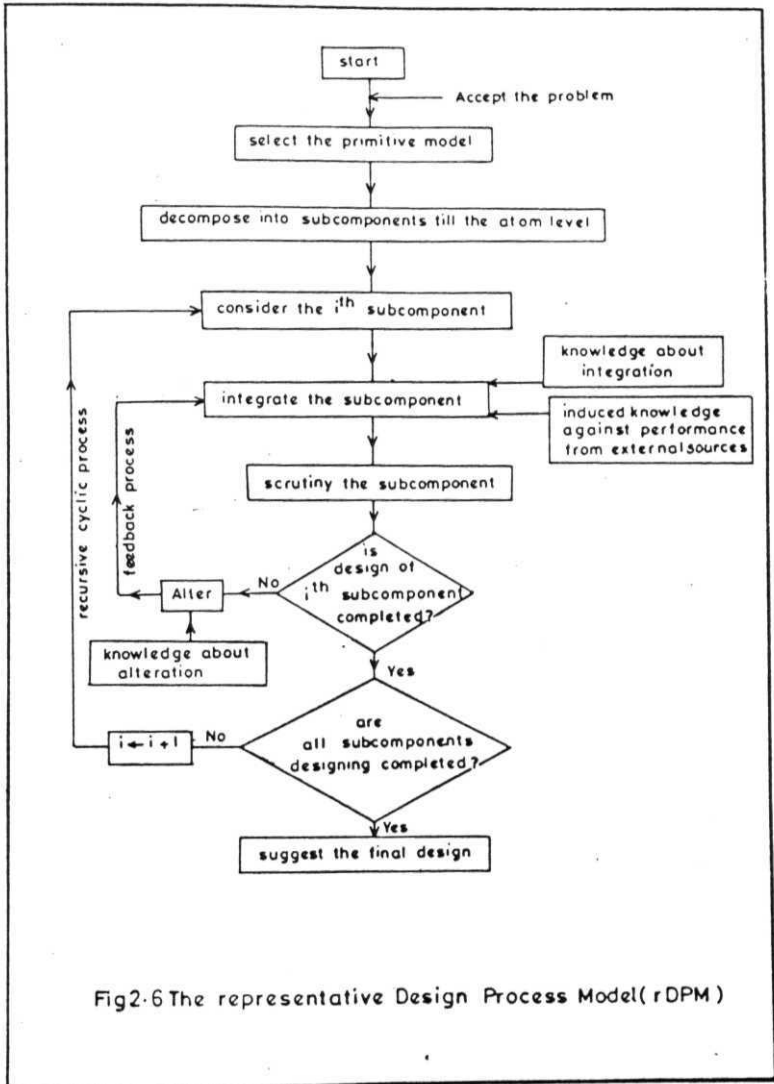


Fig2.6 The representative Design Process Model (rDPM)

## 2.6 rDPM and Knowledge Organization

In rDPM, a primitive object structure is 'selected' either from past experience or derived directly from user and is 'decomposed' further and further into subcomponents resulting in what is called component tree. The leaf nodes are **the** atom components, which are the smallest designable components and are assumed no further division though is possible is practicable. The primitive artifact(object) structure and knowledge generated out of its decomposition(obviously is knowledge of essential nature) are the sufficient and necessary arguments for design activity to proceed to any extent. The subcomponent is designed by 'integrating' the available knowledge with inducted knowledge and the resulted **subcomponent** design structure is 'scrutinized' with knowledge of analysis and 'altered' using additional knowledge of domain if necessary.

### 2.6.1 Knowledge at select Phase

As mentioned earlier, the primitive artifact structure is selected usually from past experience or from the user. Past experience is stored in the form of frames, where the knowledge of object, domain and design are stored. Each frame consists of the final object structure at that instance, how it designed the object and knowledge and parameters that are manipulated. Both successful and failure cases are stored as experience and used as episodes. At **the** time of necessity, the episode/frame that nearly matches with the current design object under consideration is selected. Here again, user is expected to interact with the system, whichever the frame matches. Another possibility is the automatic selection of system using what is called shape grammar (**Maher, 1990**).

**In** the case where there exists a model frame/episode that nearly resembles the

object under study, the design becomes obviously simple and is termed as routine design. In some cases of innovative or creative design processes, the information may be useful in pieces or chunks.

Quite often knowledge is embedded in the user either as his own experience or imagination or intuition. This knowledge can be transformed to computer as a description. In case of multiple design alternatives, knowledge is required to select one among them. rDPM assumes that it receives only one design alternative. This is one of the reasons why completely automated Intelligent CAD(Computer Aided Design) is not possible and I-CAD can provide only assistance.

### **2.6.2 Knowledge at decomposition Phase**

Decomposition is one of the critical steps of design process. Decomposition of the selected primitive structure of object leads to what is called component tree. The analysis of decomposition phase results in the generation of vast knowledge of domain, of object and of design. Since, it is the user who is disintegrating the object into components, necessarily s(he) earns the functional and relational knowledge of which and what makes the components to interact with each other. In other words, how various functions are mapped to structural attributes. In addition to the knowledge of constraints are generated in this particular step.

Classically, this process of generating knowledge out of decomposition is termed as deduced knowledge. The generated knowledge, rDPM contemplates, becomes essential knowledge for design because this is the knowledge available apriori before the beginning of design.

The knowledge of functions, attributes and constraints identified during this process is the minimum expected of decomposition and consequently designing

is supposed to be building upon this knowledge. Often, this is useful as a clue for innovative and creative design activities.

### **2.6.3 Knowledge at integrate Phase**

Integration, as was put earlier, is a phase wherein the object is constructed component by component starting from the leaf nodes of component tree. Knowledge manipulation is the essence of this step. Since, the learned knowledge while decomposition phase of what attributes constitute the given function(s) may not be completely sufficient to define that particular function(s), the additional knowledge is inducted from various knowledge resources in such a way that complete attributes set is recognized and followed.

The inducted knowledge is what rDPM called desired aspects of design. The process of design synthesis is satisfaction of essential functions completely and desired functions to the extent possible depends upon the context. In case of a trade off, desired functions are ignored to essential functions.

### **2.6.4 Knowledge at scrutiny Phase**

The recognition and amalgamation of knowledge resources leads to the designing of the component. Often, a number alternatives will be generated and selection and isolation of single alternative amongst them is a critical job. In addition, while constructing the component under consideration, it is plausible that all functional aspects cannot be satisfied completely and there may be trading off and optimization. This may give a best possible component but we are still in/dark about the consequent influences on the overall design of the object besides the attributes, constraints and knowledge required to design the component. Thus, scrutiny phase

is significant in designing process. Here, *prognostication* of influences on overall form of design object ( $K_{obj}$ ,  $K_{des}$ ) results from the deductive analysis of the **integrated** component. The outcome of the scrutiny phase is either designing the next component in the component tree (in case design is sufficient) or it makes a way for alteration or redesign the component.

### 2.6.5 Knowledge at alter Phase

If the designed component is failed in the scrutiny phase, it undergoes, alter phase. Designing an object is said to be failed if and only if

1. *The prognostication results in dissatisfaction of functional aspects of the complete object and*
2. *The component subject scrutiny is not able to fulfill the functional aspects which it supposed to.*

Alteration becomes a process of modification if designed component is nearly meeting the functions contemplated. Otherwise, it becomes a process of redesign if the design demands a change in many aspects. In other words, the component design is modified if it failed at satisfying desired functions and is redesigned if it failed at satisfying essential functions. In terms of knowledge patterns, alteration is akin to the process of **integration** (Naga Prasad & Reddy, 1992).

## 2.7 Summary

The present chapter aims at the understanding of what design process is in generic sense **and** how it progresses. The study tries to formalize the design processes, wherein, so far human brain excels. It is suggested that a further classification of function set into two subsets namely viz essential function set and desired function set makes the understanding of design task a more transparent and embodies different models proposed earlier. As a prelude, the role of design object functions, constraints and operations are discussed before attempting to model. A **Design Process Theory(DPT)** is postulated chiefly to visualize the design description. A **representative Design Process Model(rDPM)** based on this theory is proposed to deduce the final object structure from primitive object structure. It was identified that the primitive structure , though lacks many features as that of the final object, is a necessary and sufficient condition for arriving at the final goal.

The paradigm **select-decompose-integrate-scrutiny-alter** is apprehended suitable for generic design tasks. This is precisely the objective of the rDPM. The integration of knowledge of decomposition and the knowledge from induction through external sources, is reckoned with **in** the integration step which is the kernel step in the design process in order to realize the subcomponent structure.

This step is followed by scrutiny that includes even engineering analysis and if needed the designed part is subjected to alteration. This feedback **process** terminates when a subcomponent is designed. The recursive process of designing all individual subcomponents and linking them by appropriate operations, terminates when final object is designed to the expectations. It was identified that the human designer processes design both at knowledge and parametric planes. This

distinction makes human decision process easier and consistent.

The present work also identifies, the knowledge kinds involved in the generic design process and discusses the role of knowledge. It identifies that knowledge is principally of three parts: knowledge of object( $K_{obj}$ ), knowledge of domain( $K_{dom}$ ) and knowledge of design( $K_{des}$ ). Much of the knowledge is generated by the critical analysis of primitive object and is recognized essential knowledge for **the** design activity. The study suggests that identification and realization of as many desired functions as possible both at component and object level needs the inducted knowledge from environment. Further, the study suggests that the fruitful amalgamation of these two leads to 'satisfactory' design. It was shown in addition, that the three principal kinds of knowledge possesses interspersing nature and was proposed how they interact with each other. The focus of the study is only on the the role of knowledge in design process as according to rDPM. Since rDPM is generic and universal, the role of knowledge it identified **reflects** in any type of design task in general.

## cognitive Design Process Model(cDPM)

### 3.1 Introduction

In the representative model of design process based on proposed generic Design Process Theory, the questions attempted are

1. What is design ?
2. How does the design process proceed ? and
3. What will be the morphology of the design process ?

**In** addition, the propositions that control intelligent assistance **to** designer are discussed along with the paradigm of process of design. A comparative study with earlier work in this context has been carried out.

In this chapter, human epistemological activity while designing is studied and modeled. As a prelude, the brain and the computer are contrasted in light of problem solving techniques. Also, studied the limitations of both logical device(the computer) and the physical device(the brain) with reference to displaying intelligent behavior. Later, some important cognitive processes(reasoning) of the brain are highlighted which forms a basis for studying epistemological activity in design.

Usually, whenever a human is posed with a task, his first job will be to look for any equivalent or near equivalent task that was addressed earlier and how it was

confronted. Much is the same token, in the pursuit of making machines intelligent, the obvious way of starting is to investigate the presence of any such intelligent entity other than human brain.

'Intelligence' is easier to recognize than to define. However, attributes of **an** intelligent agent can be perceived (Table 3.1). If **the** attributes of intelligent behavior are examined, it is not difficult to find examples of superior animal performance in each of the attribute categories. In the objective sense, it appears that man differed **from** the **higher mammals mainly** in degree of intellectual ability rather than in having some unique and unshared capability. However, till date, human brain is considered as the only biological organism that shows entire attributes of intelligence. If so, the obvious question follows: what about **machine intelligence** or intelligent programs? Some programs do exhibit at least few attributes of intelligence. In **Schildt's (1987)** opinion, computers do routinely think and that virtually all programs exhibit some form of intelligence. He defines an intelligent program as the one which exhibits behavior similar to that of a human when confronted with a similar program. He asserts that the program need not necessarily think like a human as long as it appears to think like one.

### **3.2 Human Brain vs Computer**

Computers emerged decades ago to mechanize the process of converting raw data into information that humans could comprehend and use in decision making. Now, we are entering the era in which they will convert information into knowledge showing us how to design computers, find ore deposits and otherwise accomplish our purposes. Perhaps, a day will come, when they will begin converting knowledge to wis-

The following are the attributes of an intelligent agent:

1. Have mental attitudes such as beliefs, desires & intentions
2. Learn
3. Solve problems (by complexity reduction by decomposition)
4. Understand (including making sense out of apparently ambiguous / contrary information)
5. Plan and Predict the consequences of contemplated actions
6. Know the limits of its knowledge and abilities
7. Draw distinctions between situations despite similarities
8. Be original, synthesize new concepts and ideas and acquire and employ analogies
9. Generalize
10. Perceive and model the external world
11. Understand and use language & related symbolic tools

Table 3.1 Attributes of an Intelligent Agent

**dom - advising us as to what purposes are truly worth accomplishing -Alexander(1982).**

In a great voyage to attain intelligent behavior **in machines**, it is **imperative** to understand the architecture of two objects poised to show this behavior: **the brain and the computer.**

### **3.2.1 The Architecture of Human Brain : An Outline**

Human brain is dominated by a massive cortex which is bilaterally symmetric in structure. Each cortical hemisphere is composed of four major regions / lobes: the frontal, parietal, temporal and occipital. Every lobe is associated with **specific** function(s) although some high level functions are known to be distributed across more than one cortical lobe.

Initiative, anticipation, caution and general control and planning of actions are related to frontal lobe, whereas integration of perceptual information especially speech and vision are related to temporal lobes. On the other hand, the symbolic process such as reading, writing and arithmetic are correlated to parietal lobe. The occipital region is connected to vision. Convolutions present in frontal and temporal regions in cerebral cortex, which are 'programmed' to function after birth are associated a part(at temporal lobe) to speech and a part(the rest) to the interpretation of events in light of past experience(called interpretation cortex).

These two hemispheres are specialized to deal with problems in different ways by the use of two distinct types of paradigms. The 'sequential(or logical) paradigm' is based on a problem solving approach that considers only small portions of the available data at any given time. On the other hand, the parallel or Gestaltian **paradigm'** process data on<sup>a</sup> global basis or all-at-once. Left **hemisphere** concerns

with the former paradigm and is amenable to NLP, logical reasoning, planning and time sense. The right hemisphere is competent at tasks requiring Gestaltian synthesis such as image processing, visual and analogical reasoning and body sense and coordination (**Gazzania, 1985**). **Penfield(1978)** views the sensory interpretation and control areas of the cerebrum as a 'computer'. The structures that support the conscious awareness is located primarily in the higher brain stem and in the uncommitted areas of the cerebrum. Memory, the ability to store and recall past events, is a critical component of intelligence. The association areas of all the four lobes and part of the limbic system especially/hippocampus are thought to be involved with memory. Nevertheless, little is known about the actual storage mechanism and even less is known about their ability. Physical attributes are outlined in Table 3.2.

### 3.2.2 The Functional Architecture of Computer; An Outline

Like any other mechanical device e.g. a clock or an adding **machine** computer too exhibits patterns of activity that represent some external world. If so, why computers are expected to behave intelligently whereas others are not? The differences, **Winogard and Flores(1986)** suggest that are all differences of degree, along a number of dimensions such as

1. **Apparent autonomy** : *Clocks exhibit a kind of autonomy in  $\lambda$  operation that once it is allowed to run, it runs without human intervention. The computer exhibits this kind of autonomy to a much larger degree,*
2. **Complexity of purpose**: *Unlike many other devices, though equally complex in its construction, computers need not have a single purpose. There*

Attributes	Brain
Types of processing elements	Neurons: upto 100 varieties, functional differences not known
Number of elements	$10^{10}$ - $10^{12}$ neurons
Brain Size(volume)	1500 cc  Neuron(cell) body diameter: 0.01 cm  Axon length: in the order of 100 cm
Weight	1.2 - 1.4 Kg
Power	10 Watts
Transmission speed	ranges from 1-100 m/sec
Switching speed	$0.5 \times 10^{-3}$ sec
Interconnection complexity per computing element	upto $2 \times 10^5$ connection(for Purkinje cell)
Component reliability	low, neurons die continuously

Table 3.2. Physical Attributes of the Human Brain

*is no way to assign a single global purpose to any one detail of its construction or operation. On the other hand, each detail may be the result of an evolved' compromise between many conflicting demands.*

3. **Structural plasticity:** *In most physical mechanisms, their dynamic changes(the motion of the parts) do not change the basic structure and hence, the basic structure remains same over time. However, in the case of computers, every time a program is run, a new one is added, the system undergoes a change that may cause it to act differently in future.*
4. **Unpredictability:** *As a consequence of complexity and plasticity, the activity of the computer is often unpredictable even when running without errors. In interacting with such a system, treatment tend to follow in the same way, the organisms ore treated - perturb and then understand the reaction.*

The idea of converting an inanimate object into a human-like thinking activity is as old as mythology. In 1650, Thomas Hobbes proposed the idea that thinking is a rule based computational process. Since then many (**Lebnitz, 1661; Boole, 1854; Russell & Whitehead, 1910**) were under the opinion that a formal system for reasoning is possible. **Turing(1950), Neumann(1956), Barrett(1979), Simon(1987)** discussed the limitations and possibilities of mechanization of thought.

Digital computer, the most widely used form of the computer is an instruction following sequential symbolic processor. Computer hardware is controlled by an internal program called operating system. Nevertheless, for the user, the separation between hardware and operating system is invisible. Their combination

forms the computer which 'understands' instructions presented in one or more specialized languages.

Procedures are described to computers by means of **programming languages** which have precise syntax and use. The languages, unlike natural languages, are void of ambiguity and context dependency and are well-formed. Procedures reflect what user wants computer to do besides how it has to do(procedural instruction).

Data is represented isomorphically or symbolically in computers. In former representation, data is modeled by a quantity which has a functional resemblance to the original data itself. This type of representation at present is used in analog computers and hence are fast but is limited in accuracy and flexibility. On the other hand, in the later representation, data is completely independent of the characteristics of the data being represented. This type of representation is used in digital computers, hence is highly flexible and accurate but is relatively slow.

In the way of introspecting about intelligence, the architectures that produce functional behaviors of intelligent animate device(brain) and to be emanated inanimate device(computer) are contrasted. It is almost concluded that/computer is the only logical device, that can be considered to fulfill the desire of making **human-like** intelligent machine.

### **3.2.3 Limitations of Brain and Computer**

**In 1931, Godel(Fischler et al, 1987)** as against to the belief that all problems can be formalized and can be solved by machines, proposed that some problems cannot be solved by computers(e.g. Hilbert's tenth problem, Halting problem and Busy beaver problem). **Fischler & Firschain(1987)** consuirmed the implications of Godel's theorem as:

1. **Man and machine limitation:** *Both brain and computer consists of 'hardware' that operates according to strict mechanical laws. Therefore, there will be truths unknowable by both brain and computer.*
2. **Only machine limitation:** *Brain exceed strictly mechanical limits by their ability to introspect and to interpret experience.*
3. **Neither man nor machine limitation:** *Most of our knowledge comes from inductive methods that operate in inconsistent systems. Godel's theorem simply places a limit on one mode of obtaining new knowledge.*

In addition to unsolvables(Godel incomplete theorem), intrinsically difficult **problems** too (Fischler et al, 1987), pose computational complexity. These problems are characterized by the fact that their solution time grows exponentially with some **parameter** indicative of problem size.

Coming to the case of brain, the problem is with their unreliable **components**. Below certain level of complexity, any system tends to break down and become random. However, there exists **such** an organization, in case of very complex systems, that despite the loss of some individual components, they continue to function. Needless to say, brain is of one example. A **statistic** estimate that everyday/brain loses upto 1000 neurons and are not replaced. **Neumann(1956)** hypothesized that neurons can be thought /as logical switches, whose massive redundancy results in an arbitrary degree of failure tolerant **operation**. However, this model does not provide a mechanism for compensation, **regeneration or** evolutionary improvement.

As a summary, it can be stated the views of Simon(1987): **at** the root of intelligence are symbols, with their derogative power and their susceptibility to manipulation. Such symbol manipulation can be done either by man or by any computer. In other words, there is no reason why machines cannot be capable of human-like understanding of external world.

### 3.3 Cognition and Design

The word **cognition** or cognitive activity pertains to the human's reasoning and problem solving methodology. They reflect a more introspective approach. **Getmanova(1989)** observed three forms of human thought: concepts, judgments and inferences. She defined 'concept' as a reflection of concrete objects and their properties using forms of sensory cognition - sensation, perception and representation. Judgment is a form of thought in which something is affirmed or denied about the existence of objects about the link or relations between an object and another. An inference is a form of thought which derives from one or several judgments, a new judgment with a certain degree of probability from 0 to 1.

The structure of inference contains premises, a conclusion and a logical relation between the premises and the conclusion. Inference, in this context can be put as the logical transition from the premises to the conclusion. What is designed in the perspective of cognition is the observation of various stages of design activity and the ways of manipulation of knowledge.

There are three major principal types of inferences: **deductive, inductive and abductive**, although there prevails certain inferences which involve more than one inference mechanism. Abduction is a kind of inference which under given premises results in a conclusion that is a sort of explanation. In other words, abduction is a process of generating explanations (**Charniak & McDermott, 1985**). They suggested the paradigm

From : b

(if a,b)

infer : a

In the same way **Rich & Knight(1991)** described abduction as

'given two **wff**'s(well-formed formulae or prepositions)

(A  $\longrightarrow$  B) and (B)

for any expressions A and B, if it is consistent to assume A, do so'.

In diagnostic domain, **Gero(1987)** quoted from (**Peirce, 1878**) that abduction is a process of generating hypothesis and showed two types of abductive diagnostic steps: non-contextual and contextual. In non-contextual steps, a hypothesis is generated by observed 'findings' and in contextual steps, a hypothesis is generated in the context defined by another hypothesis.

Deductive inference is a well established mechanism in AI. Hence, a definition is thought suffice to illustrate: deductive inference is one in which the conclusion necessarily follows from the premises which express knowledge with a higher degree of generality, and itself represents an item of knowledge with a lesser degree of generality.

Inductive inference arise as a form of generalization from the judgments or premises of lower generality to a higher piece of generality. Inductive inference is of two types, complete induction, where premises are universal set of facts about the class of object under consideration and incomplete induction, when the observed facts are limited. Incomplete induction is further divided into three classes; popular induction, induction by analysis and popular scientific induction of causality (**Getmanova, 1989**). She suggested that the recurrence of one and the same property in a series of homogeneous objects, and the absence of any

examples to contradict this, serve as the basis for the conclusion that all objects of the said class possesses this property and is called popular induction. The second type is slightly differed from the first one in the sense that there is no restriction of homogeneity in observing objects. Inference in which cognition of the essential features or the essential relationship of a part of a class of objects is used to obtain a general conclusion in scientific induction of causality. It is observed that all these three types of principal inference mechanisms are used in design.

Before studying the cognitive processes of design, the aspects of general problem solving will be discussed in this chapter, with emphasis on reasoning mechanism. Later, the design process is looked into from a cognitive point of view. As a first step, the protocols of various domains are analyzed. Next discussed was, how others model the cognition of design. Finally, the cognitive Design Process Model(cDPM) is proposed.

### **3.3.1 Protocols in Design of Various Domains**

Domains like chemical, civil, mechanical engineering etc traditionally do involve design processes. Analysis of their protocols infers that various phases of design and reasoning methodology involved are under analysis.

#### **3.3.1.1 Design of a Process Heat Exchanger**

Bell(1983) suggested the following steps in design of a process heat exchanger

1. *Process conditions must be specified(stream composition, Flow rates, temperatures and pressures).*
2. *Required physical properties over the temperature and pressure ranges of*

*interest must be obtained.*

3. *The type of heat exchanger to be employed is chosen.*
4. *A preliminary estimate of the size of the exchanger is made, using a heat transfer coefficient appropriate to the fluids, the process and the equipment.*
5. *A first design is chosen, complete in all details necessary to carry **out** the design calculations.*
6. *The design chosen in step (5) is evaluated against expectations with respect to both heat transfer and pressure drop.*
7. *Based on the result of step (6) a new configuration is chosen if necessary and step (6) is repeated.*
8. *The final design should meet process requirements at lowest cost.*

With the aid of knowledge of earlier discussions, it can be said that the procedure **Bell(1983)** suggested is of routine kind of design. Depending on the knowledge of selection criterion, a design alternative is chosen and detailed design is conducted. The object is scrutinized against expectations and if necessary modified or another alternative is chosen. This process is repeated till the goal is satisfied.

### **3.3.1.2 Design in Structural Engineering**

**Fennis** and **Novabhoompipat(1978)** consummated that structural engineering design generally consists of three phases:

1. *Preliminary design in which a structural configuration is selected and preliminary component parameters are chosen*
2. *Analysis in which the response of the structure is evaluated and*
3. *Detailed design, in which components and connections are selected, so as to satisfy the design objectives and the requirements of applicable design specification and building codes. They identified that analysis and detail design are frequently iterated until convergence is reached.*

**Periera(1978)** while discussing about the utility of AI techniques in layout design, discussed the phases of design. Given list of imposed and non-permissible adjacencies among a set of planar rectangle spaces and given dimensional intervals, the design problem in layout design is to generate all possible layouts which satisfy the requisites. He identified four phases of design: problem representation and definition; problem decomposition into modular subproblems; testing problem data well-formedness and in case of necessity, modification.

### **3.3.1.3 Electrical Circuit Design**

**McDermott(1978)** observed that design might be described as a process of adding detail to **description** until 'full detail design' is reached relative to some basis. The initial description enunciates what the device is required to do; the final description furnishes the structure of the device. He proposed that the basis of design domain is a set of primitive devices that have no internal structure, whose elaboration consists mainly of selecting values for their control attributes.

During the initial phase of circuit design, the components of the device of the type are defined. This is followed by establishing connections and constraints

between components. Design in data terms start next. The control attributes are given with numerical values on which, the author argues the designer has complete or partial control over. The design of components leads to partial solutions and later, into complete solution by what is called successive refinement. Errors will be **addressed** later by execution of error-connection plans.

**Sussman(1978)** proposed **SLICES** for circuit design : a descriptive mechanism of an automatic analysis procedure. In this context, the problem of engineering design described by him as, given a starting set of devices with known behavior and set of rules by which these may be combined to produce more complex entities, construct a composite mechanism, whose behavior satisfies certain specified properties.

He mentioned that for each design problem, engineer must establish **the form of the answer**. If the problem is known, the engineer retrieves several competent forms. In most engineering domains, he pointed out that the form of the answer is the topology for a mechanism with many undetermined parameters. In case, design problem is not familiar, it must either be reformatted into a familiar problem or decomposed into a combination of more familiar problems. The composition of solution to subproblems may lead to unforeseen interactions requiring debugging.

He identified that the kernel of the engineering design is a plan that illustrates the mechanism being designed at many levels of detail. According to him, **in** synthesis part of the design, the description of the parts are inter-connected to form the whole. Nevertheless, in analysis part, the aspects of the resulted behavior from its individual parts are determined. The plan includes not only a description of the physical connection between the parts **of** a mechanism, but also a description of how the behaviors of the parts interact and constrain each other to produce the

overall desired behavior.

**Preiss(1978)** viewed design process as involving a series of steps, that can interact with every other step. However, he pointed out that a good designer takes decision in such a way that these interactions are minimized. Fig. 3.1 shows schematically the steps of design process. He presented that the product to be designed is considered to be the goal node in a hierarchy, which he termed abstract hierarchy tree. At this level, only those parameters required to define the object with its uses are defined. He argued, that it is not necessary to define all the properties of all the subsystems at this step. The goal node, or product is divided into subsystems, and so on, until the lowest tips of the tree are reached which are the primitive components used or made for assembly into each subassembly. Any node in the tree can interact with any other node. Division of the elements of data into each hierarchic node is part of the design process. Design proceeds by instantiating variables in the nodes by constraint propagation. Constraints may be from inferences or from computation.

In summary, what the protocol analysis of individual domains expressed is that design starts with getting conceptual idea over the overall form of the object to be designed followed by its conceptual analysis and synthesis. By and large, one should design the object at conceptual level and form a plan about the design. This inturn to be executed at data level. If any constraint(s) failed, analysis schema has to be evolved and the first design is modified / redesigned.

### **3.3.2 The Role of Analysis and Synthesis**

The words analysis and synthesis are extensively used throughout this thesis. Objects are 'synthesized' by bringing knowledge from many subobjects. More

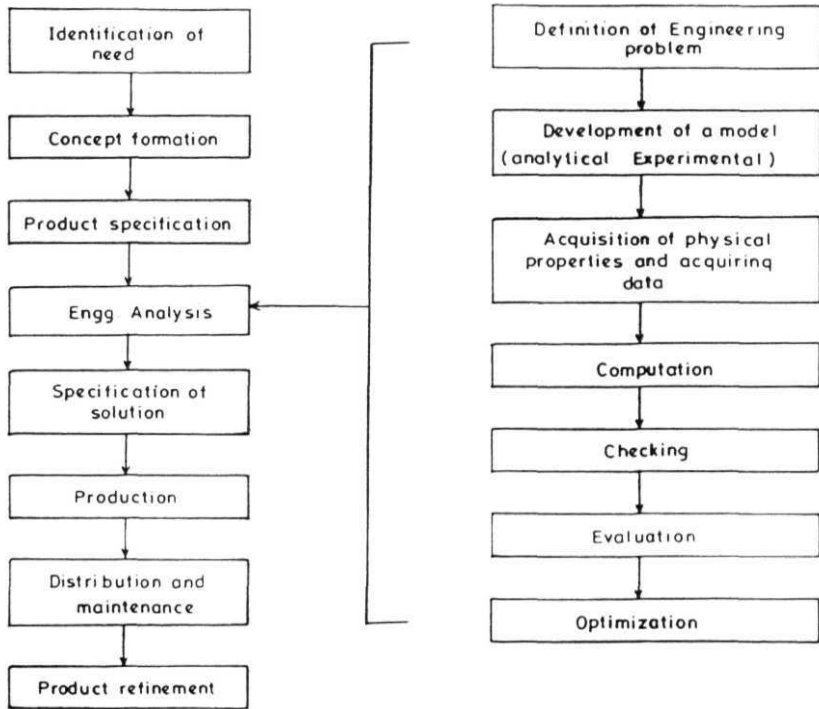


Fig.3.1 The steps of design

general concepts are synthesized from less general concepts. That is, for anything to synthesize, it requires that

1. The goal and *subgoals*(*object, subobjects; more general concept, less general concepts*) *should exist* and
2. *There must be structural or conceptual relations that should exist between subgoals.*

As a consequence, the synthesized object shows different behavior from that of the individual components.

Eastman(1978) defines synthesis as the generation of new configurations so as to satisfy earlier defined functions. He pointed out that, if synthesis is found to be difficult for one subtask, it is often iterated at a higher level in order to reframe the lower subtask. Synthesis is often a non-deterministic process and involves a 'search' for an acceptable solution. Some synthesis processes do not involve search, but are procedural methods. He suggested that as long as procedural methods derived from simple deterministic relations are available, the synthesis can go without analysis.

In design, the object is analyzed in order to understand its behavior and the attributes that cause the behavior. The analysis consists of deriving new concepts for an entity by applying a model to either other attributes of the same entity or to the attributes of others that comprise the entity (Eastman, 1978). Sussman, 1978) in different context suggested the same and proposed that, analysis moves information up in the design plan by determining behavior of the composite mechanism from appropriate aspects of the behavior of the parts and their

relations.

### **3.3.2.1 Relations between Analysis and Synthesis**

**Sussman(1978)** showed that analysis and synthesis go hand in hand in design. Both analysis and synthesis processes rely on relations within the abstraction hierarchy, according to **Eastman**. The breadth of relation in the hierarchy encompassed by one analysis sometimes encompasses many entity descriptions also. However, these cases are only a few. Nevertheless, one can classify all generic task solutions as either significantly analytic(e.g. diagnosis) or significantly synthetic(e.g. planning). The word significant is very important since no task solution in universe is completely analytic or completely synthetic (**Wolf, 1985**). In synthesis dominated problems, the brain employs analysis for assessing, whether the observed behavior of synthesis is the one which is expected behavior and what best have to be done in order to bring it to the expectations. Similarly, it can be argued for deductive jobs. **Gero(1987)**, while trying to find a generalized model for diagnostic models(primarily analytic), suggest that both inductive and deductive logics have to be exploited in diagnosis tasks.

### **3.3.3 Cognitive Modeling of Design Process: Previous Work**

As previously discussed, human being designs an object to change the world in which he is living with a specific motive. In all design tasks, certain similarities, especially in epistemological activity, are observed. The epistemological process is independent of the domain, but is dependent on the generic task type(here design). Studying cognitive process behind design process has atleast two advantages: One, it makes to know what made the brain superior, and two, it paves the way to appreciate its techniques in addition to formalize and represent them explicitly.

There are two approaches that are discovered so far, to study cognitive processes in design. The first approach tries to explicate the definitions, typologies and prescriptions of the human discourse which are inferred implicitly in him. The later approach lays stress on the human cognitive hardware such as notion of memory and network types. The former approach is called classical cognitivism and the later is called connectionism (Coyne, 1990). Both have their own axioms of applicability and it is difficult atleast at present to contrast their relative advantages. Nevertheless, the present study operates under the premise that the epistemological activity follows certain formalism and it is possible under certain limits of scope to explicate and hence adopts classical cognitivism. Takeda et al(1990), Gero(1990), Maher(1990), Chandrasekharan(1990) are a few who followed this kind of study.

Takeda et al(1990) distinguished two levels in the design process when considering the designer's mental activity. One is at the object level, where the designer thinks about design objects themselves, for example, what properties the design object has and how it behaves in a certain condition. The other is the action level, where the designer thinks about how to proceed with his(her) design, that is what he/she should do next. The designer seems to perform his(her) design mutually using these two types of thinking. When looking at the design cycles with respect to this aspect, they also contain these two levels(Fig. 3.2).

In logically describing design processes, they proposed that it is reasonable to assume the following simple model as a first step;

$$S \cup K \vdash D_s$$

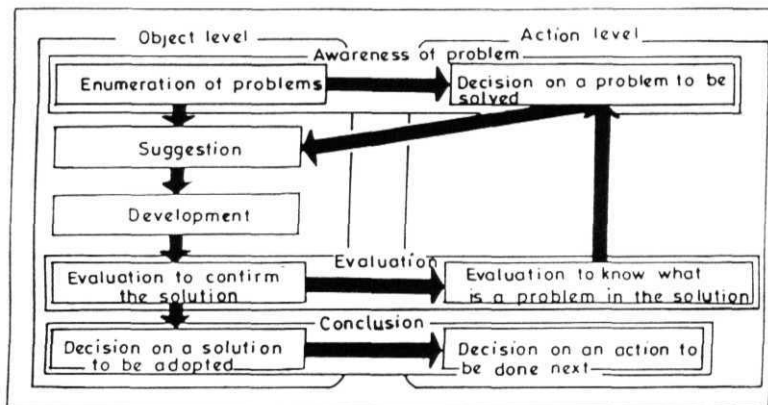


Fig 3:2 Design cycle

Where S,K, and **Ds** are a set of formulae that denote the specifications, knowledge used in design, and the design solutions respectively. The solutions are derived from the specifications and knowledge as a result of deduction. However, this **formalization** has three difficulties in interpreting the whole of the design process. First, design is not always performed with complete information, which implies that refinement of specifications and designing of objects are performed **simultaneously**, . They not only use deduction but also abduction for **the** formalization of design processes to **refine** the specifications. Second, **knowledge** in the previous assumption is concerned with how to design objects. A typical example is, 'If there is a specification **S1**, then use a design object **D1**'. Although, it might be useful for routine design, it is not appropriate for more flexible and creative designs in which knowledge about object properties and behaviors play an important role. The third problem, which is related to the second, is knowledge incompleteness and inconsistency. When the designed object does not satisfy the specifications, the knowledge base should be regarded as initially **incomplete** rather than inconsistent.

In this context, design solutions and knowledge will be the premises and the specification as the conclusion. Based on this assumption, design processes are formalized as bidirectional and iterative processes as follows :

$$\begin{array}{c} \overbrace{D_s \cup K_o \vdash P} \\ \underbrace{D_s \cup K_o \vdash P} \end{array}$$

Where  $D_s$  is a set of logical formulae describing a design candidate,  $K_o$  is knowledge of object properties and behavior, and  $P$  is the set of properties of the design candidate. Required specifications are included in  $P$ . Given, the design knowledge  $K_o$  and the required properties  $P$  as the specifications, the designer tries to find a candidate. Deduction is then performed to see (1) what properties the

candidate has and (2) whether the candidate contradicts the given constraints, including the specifications. The result is that the description of the obtained candidate becomes complete by detailing. If the candidate does not satisfy the specifications, the designer either tries an alternative candidate or modifies the design knowledge and the specifications. Trying an alternative candidate recalls further abductive or deductive processes, and finally, a complete description of the solution with the specifications are obtained. If there is no other way to explore, the design process terminates.

The previous formalism solves the first and the second problems but not the third one, which is related to the incompleteness of knowledge bases. Circumscription is introduced for this problem and assumed that every piece of knowledge is valid only when it is used in certain situations. However, it can only identify the applicability of knowledge when detecting a contradiction. For a given set of logical formulae, circumscription can be used to compute exceptions that caused the contradiction. By doing so, the original knowledge is modified, and it is able to handle the incompleteness.

They also proposed the idea of employing three different types of reasoning: deduction, abduction, and circumscription, for formalizing design processes. Here, computable model of design processes has shown by them based on these three types of reasoning and interpreted the cognitive model in the framework of the computable model.

Assuming that a design process changes its state step by step, the following formulae holds according to this model

$$Ds_c \cup Ko_c \vdash P_c$$

where  $Ds_c, Ko_c$  and  $P_c$  are the description of the current design candidate, the knowledge available at the current state and the properties of the current design candidates respectively. Now, a computable model can be made by interpreting the cognitive model in the logical framework discussed earlier.

During the suggestion subprocess, the designer tries to find a feasible candidate. The purpose of this subprocess is to obtain  $Ds_c$  from  $P_c$  and  $Ko_c$ ; it can be regarded as an abduction process. Both the development and evaluation processes are regarded as deduction. In these subprocesses, the designer applies his/her knowledge to the candidates and obtains, what is known at the current state. The difference between these two subprocesses lies in the kind of knowledge that is applied. The development subprocess uses knowledge to find out, what properties the design object has, whereas, the evaluation subprocess uses knowledge to compare those properties obtained in the development subprocess with expectations. The purpose of these two processes is to obtain  $P_c$  from  $Ds_c$  and  $Ko_c$ :

$$Ds_c \cup Ko_c \vdash P_c$$

$P_c$  can be different from  $P_c$ , if and only if  $Ds_c$  or  $Ko_c$  changes.  $Ds_c$  changes as the result of abduction and  $Ko_c$  changes as the result of circumscription.

While the designer is developing or evaluating, s/he sometimes encounters a difficulty about the candidate and defines a new problem to solve it. It is a jump from a development or evaluation subprocess to an awareness-of-problem subprocess and interpreted it as circumscription.

The continuation of the design process can be disturbed for two reasons, one is due to lack of information, which is dealt with by metalevel reasoning. The other is because of a contradiction in the theory that happens not because the knowledge

contains false information but because it is incomplete. This contradiction can be avoided by defining exceptions for these pieces of knowledge using circumscription.

If a contradiction is detected in the theory, then the relevant formulae are gathered which caused the contradiction. The literals composed of abnormal predicates are added to them and these abnormal predicates are circumscribed with the theory. After words, modified formulae are obtained where abnormal predicates are substituted by non-empty formulae. They argued that, as a result, the contradiction is removed from the theory. Because the formulae are modified, it cannot always derive the formulae that represent the required specifications from the current design candidate. Some new formulae must be added to the design candidate to satisfy the specifications. In this process, the contradiction creates a new problem, that is the awareness-of-process subprocess.

In the action level of the logical framework, metalevel operations are introduced to deal with the problem. In this approach, operations on building theories and on how to perform reasoning are introduced as metalevel operations, including setting up of Ds; setting up of P; setting up of Ko; revision of Ds by abduction on P and Ko; revision of P by deduction on Ds and Ko; and revision Ko by circumscription on Ds, P, and Ko.

Starting the design cycle is interpreted by them as executing the setting up of P and the setting up of Ko operations. Similarly, it can be interpreted that the suggestion, the development, and the evaluation subprocesses as the executions of these operations. Knowledge about how to design can be represented as a sequence of operations at this level.

Reasoning on the action level is to obtain a sequence of the operations on the

object level. This reasoning is performed by using the status of the object level and knowledge about how to design. The status of the object level is determined by what kind of information is obtained on the object level and how this information is obtained. Knowledge about how to design is used only on this level and includes procedural knowledge about how to proceed with the design, the design strategy, and so on. It is possible to discuss reasoning on the action level in the same way as done for the object level. Actually, however, only deductive reasoning, such as rule-based reasoning, is needed because most of this type of knowledge is procedural according to them.

They did not interpret the conclusion subprocess within the logical framework. It is a decision-making process that considers all information obtained by other types of reasoning and therefore, it seems to be reasoning on a higher level than those types of reasoning. They left this subprocess unformalized. Fig. 3.3 summarizes the discussion and shows the correspondence between the subprocesses and the types of reasoning in the logical framework.

In **Maher's(1990)** view, in generalizing design problem solving, three distinct models of design synthesis are identified: decomposition, case-based reasoning, and transformation rules. These models are distinct in their associated formalism for design knowledge. The models are not necessarily cognitive models; although they might match various approaches humans take when producing design solutions, the correspondence has not been adequately tested. The distinction among the models, lies in the representation of design knowledge rather than in their appropriateness for a specific design domain or phase of design. The purpose of identifying more than one design model according to Maher is in identifying appropriate formalisms for representing design knowledge.

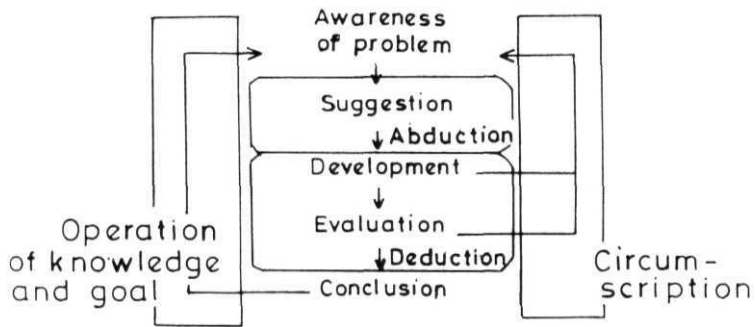


Fig.3 3 Reasoning on Design Cycle

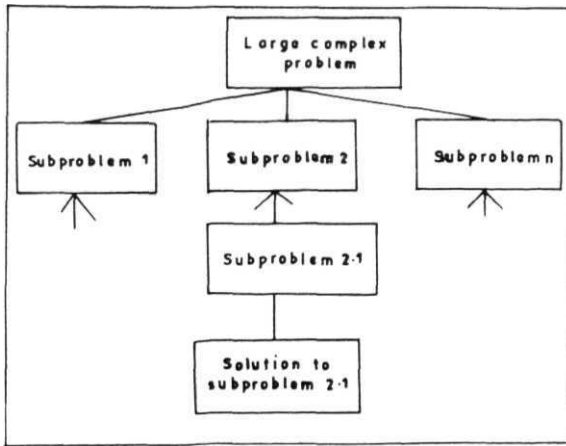


Fig.3.4 Decomposition Model

She pointed out that the decomposition Fig. 3.4) is perhaps the most ubiquitous model of design. It follows directly from the development of design methodology. The idea of dividing large complex problems into smaller, less complex problems is well accepted and she asserted that it is possible to consider all models of design as some form of decomposition. Specific knowledge based systems for design by decomposition have been developed that identify specific languages for describing design knowledge. Examples of such languages include **DSPL**(**Brown & Chandrasekaran, 1985**), **Edesyn**(**Maher, 1988**), and **Vexed**(**Steinberg, 1987**) . The issues associated with this model, include the appropriateness of decomposing and assuming that solutions to loosely coupled subproblems will combine to form a good design solution and the ease of specifying design knowledge in the domain, as decomposition and recomposition knowledge.

Case-based reasoning(Fig. 3.5), **Maher(1990)** recognized as a **model** of design, that directly uses design experience **in** the form of episodes rather than by compiling and generalizing it. The model uses analogic reasoning **to** select and transform specific solutions to previous design problems **to be** appropriate as solutions for a new design problem. This model is attractive as she describes because the knowledge acquisition for developing generalized representations of design knowledge in a particular domain may be difficult and time consuming. The issues associated with using this model of design include **the** identification of the necessary information about design episode, about its **applicability in a** different context, the meaning of similar design, and the transformation of **the solution** from the original context to the new context. Although, human designers **appear** to be good at using this type of analogy, it is difficult to automate it.

Transformation, Mahe(1990) suggested that is a holistic approach to design, similar to case-based reasoning, but uses generalizations rather than specific episodes, like decomposition. In the transformation model(Fig. 3.6) **the** design knowledge is expressed as a set of transformational rules in which the **left-hand** side(LHS) of the rule is replaced by the right-hand side(RHS) of **the rule**. The most common application of transformation model is manifested as **grammars**. The issues associated with using the model are the representation of the design description, the control in selecting an eligible transformational rule, and the termination of the application of rules.

Interestingly, **Winograd & Fenros(1986)** discussed cognition in design process in a different way. Their addressing design is on more fundamental insight; the ontological insight, where sociological and **philosophical implications** and understanding of design are presented. They proposed the following steps **while**

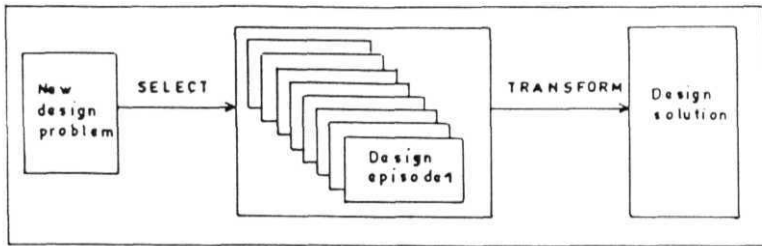


Fig. 3.5 Case-Based Reasoning Model

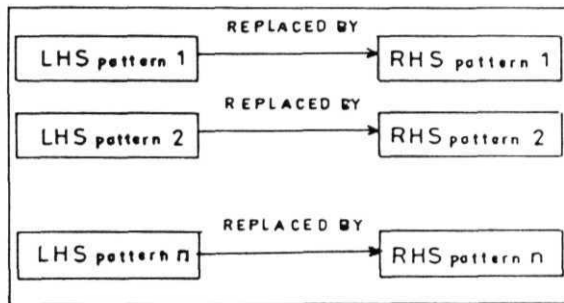


Fig. 3.6 Transformation model

designing.

1. *There is no clear problem to be solved; actions need to be taken in a situation of irresolution.*
2. *A business (like any organization) is constituted as a network of recurring conversations.*
3. *Conversations are linked in regular patterns of triggering and breakdown.*
4. *In creating tools, new conversations and connections are getting designed.*
5. *Design includes the generation of new possibilities.*
6. *Domains are generated by the space of potential breakdown of action.*
7. *Breakdown is an interpretation - everything exists as interpretation within a background.*
8. *Domains of anticipation are incomplete.*
9. *Computers are tools for conducting the network of conversations.*
10. *Innovations have their own domains of breakdown.*
11. *Design is always already happening.*

This way of understanding is something different from what either the previously described work or to be proposed cDPM confesses their way. In these studies, there is an implicit assumption in what is meant by cognitive models. They not only described psychological procedures but also are nearer to computation. The study of previous work can not be said completed if the other approach, the connectionistic approach is outlined.

The idea of connectionistic approach to design is coined by **Coyne(1990)** . Connectionism is concerned with modeling design reasoning at a low level in an attempt to replicate the capability of human reasoning to transcend the structures of sharply defined categories and formal logic. Concerning design, **connectionism** emphasize on experience and the emergence of design ideas without **attempts** to articulate the process, whereas, classical **cognitivism** in which the present study is based on supports and emphasize on design rules, **hierarchies** of types and the articulation of the design process. The connectionist models rely heavily on the notion of memory and the belief that decision emerge from this memory **in** response to some situation.

A major assumption behind this approach is that aspects of extremely large and complex systems, such as brain, can be studied in a scaled down version. The connectionist approach involves the construction of computer **models** that consists of 'networks' much similar to neural networks. A good description of neural network approach is given by **Rich and Knight(1991)**, **Rumelhard and McClelland(1987)** and with reference to design is given by **Coyne and Postmus(1990)** . However, the establishment of this approach is **still** in infant stage.

### **3.3.4 cognitive Design Process Model(cDPM)**

The unique motive behind building expert systems or in a broader sense intelligent machines is to emulate/human brain. Computers already proved their edge in the aspects such as storing vast quantities of information. They retrieve information in nanoseconds and perform extensive arithmetic operations without error. However, they are inferior in the situations where human brain excels. Brain excels in reasoning methodology even when the knowledge is **partial** and fuzzy. It

has commonsense. Imparting these kinds of capabilities to computers leads to intelligent machines which will have hopefully the best of computers as well as the brain.

cDPM, which the present study proposes, accepts the paradigm **select-decompose-integrate-scrutiny-alter** as phenomenological in the human cognitive process (Naga Prasad, Reddy & Kanakaraju, 1993). In fact, this paradigm is a computable version of what is going on in human brain. As stated earlier, design activity is performed at two interspersing planes, the knowledge plane and the parametric plane. Obviously, the paradigm functions at both planes. In cognition, **select** is an important phase idiosyncratic to human brain. Here the problem before the human designer is appraised. He brings to mind the necessary overall form of the object called primitive form that satisfies hypothetically atleast a part of the objectives of design.

**The primitive form is selected abductively.** In case of routine design, human tries to select among the alternative design forms and selects the one which is most suitable to the context. The case-based reasoning is found more valid in selection phase as the 'past experience' or 'awareness' about object apriori is an indispensable need for design to proceed.

How to select a best case which is nearer to the present context? One way is to match the functions that the case fulfills and the functions the object in context desires. These matching can be identically same(perfect match) in the case of routine design or can be partial (partial match) in the case of innovative design. **Kolodner(1989)** identified six kinds of preference heuristics among which two kinds viz salient feature preference and specificity preference are found akin to cognitive activity at select phase.

The primitive form of the object that was selected now is decomposed into subcomponents further and further. The idea behind it is to reduce the complexity. The decomposition takes place in two ways, when the design leads to structure. The decomposition is physical leading to physical subcomponents in first way and in other way, it is functional, when a component is identified as a cluster of components, who contribute to satisfying atleast one function of the object. This raises a question as which way is to be adopted and when. One approach is to ignore the distinction between function and structure. Another approach is to have separate representations for function and structure. However, it was observed that decomposition based on functional satisfaction is phenomenological. Here, the design object is viewed as a logical ensemble and its chunking leads to logical entities and further into logical elements. These are defined by atleast a logical principle and connected by logical bridges / connectors. A logical principle is the function that the object has to fulfill and a logical bridge is the relation between two structural elements each can independently satisfy some functions **stipulated** to them among which few are common. Now, the result of decomposition based on functions is mapped to structure, since, the element that satisfy particular function can also possess a particular physical structure. **The process of decomposition is deductive** because the conclusion necessarily follows from the premises.

This primary treatment of hypothetical design object provides the designer in identifying

1. *the components or elements, the object 'basically' constitute of,*
2. *the relations(logical / structural) between them and*
3. *the operation(logical / physical)*

involved in connecting two or more elements to transform into its parent component. **From** now onwards, both the structural and logical description of object's <sup>the</sup> **components**, elements and operations are treated as fundamentally/same. No difference between them is contemplated.

After decomposition, human being is aware of the subcomponents, the relations between them to configure the **object**. This is the **minimum** knowledge required to synthesize an object and is essential. This is sufficient for routine design, whereas, it is the basis for innovative or creative designs.

The process of synthesizing the elements starts now, which is **the** heart of the task objective's realization. The integration of knowledge arised out of decomposition phase and the knowledge from induction through external sources, is reckoned within the integrate step to realize the subcomponent structure. The first step in integration is to find hierarchy among elements to be designed for a fruitful amalgamation. Now, the design is worked out in parametric plane, such that, it results in structural description of the parent subcomponent. In case of non-routine designs, the user asks himself, whether he can improve the subcomponent by fulfilling some of those constraints which are not attended earlier due to lack of knowledge or ignorance or any other unforeseen reasons. Intention of addressing constraints in a better way than earlier is a seed to innovation or creation. ***K (Knowledge of essential aspects) is brought to the present context abductively. K (Knowledge of desired aspects) and K (Knowledge of essential aspects) are integrated inductively*** to produce a provisional model of subcomponent. Needless to say, it need not have to be exactly fit into the expected behavior. Hence, it undergoes scrutiny. Like this one after the other, all subcomponents are designed.

For example, consider the following rule type in the context of designing the hairpin heat exchanger. Consider, the design of subcomponent inner pipe.

### **SHELLSIDE**

IF

(Fluid is liquid)

(Its viscosity is high)

THEN

(It requires large heat transfer surface area A,)

Large A,(surface area) is attained by increasing length of the each tube. However, large A, can also be provided by fins on the pipe. Abductive inference for getting finned tubes is used in the above condition.

Abduction leads to expectations and often, they may or may not be completely true. For example, if the fluid flows on the fins is corrosive(forms scales **quickly** thus needs cleaning at a faster rate than bare pipe) then the alternative of finned tubes may not be viable economically.

The scrutiny phase is similar to a classical diagnosis problem(Gero, 1987). The input for this phase is the designed part of the object, which is a sort of hypothesis to the diagnosis task. The designed object is subjected to scrutiny against fulfilling the local(limited to subcomponent) and global(extended to **complete** object) functional aspects. This step is deductive. At every stage of building subcomponents, deductive mechanism is applied to qualify the objectives of subcomponents(**assuming** that every subcomponent has its own specification called objectives). **Alteration** is required, when the subcomponent in context has not

accomplished the expected behavior. Scrutiny phase will give sufficient information to the human designer to take a decision whether subcomponent is to be modified or it needs a redesign. The process of alter again is contemplated similar to integrate phase.

While altering, much of the knowledge is inducted in order to improve the component under design. Its role is quite big especially in innovative and creative designs. The new ideas will be implemented in these cases.

Fig. 3.7 depicts comprehensively the epistemological activity behind human designer. cDPM helps in understanding design and constructing AI based systems for design. Chiefly, it contributes to learning what type of assistance or intelligent assistance, the human designer demands at various phases of design from intelligent CAD systems.

### **3.4 Summary**

*The* brain is an enigmatic organ, which shows simultaneously many attributes of intelligence. Study of architecture of brain may not be much useful atleast in the context of arriving at intelligence in machines, the way brain achieved. However, it has one purpose: one can understand what is meant by intelligence and how to solve problems in non-procedural contexts. Hence, it is not required to replicate human behavior in the objective of realization of intelligent computers but rather suffices to exhibit intelligence by machines. This automatically suggests to analyze the protocols of the domains with reference to problem solving.

In this chapter, the brain and computer architectures are analyzed to see whether computer can be made intelligent at all. Later, in the context of design, different domain protocols are analyzed. Two important conclusions were drawn;

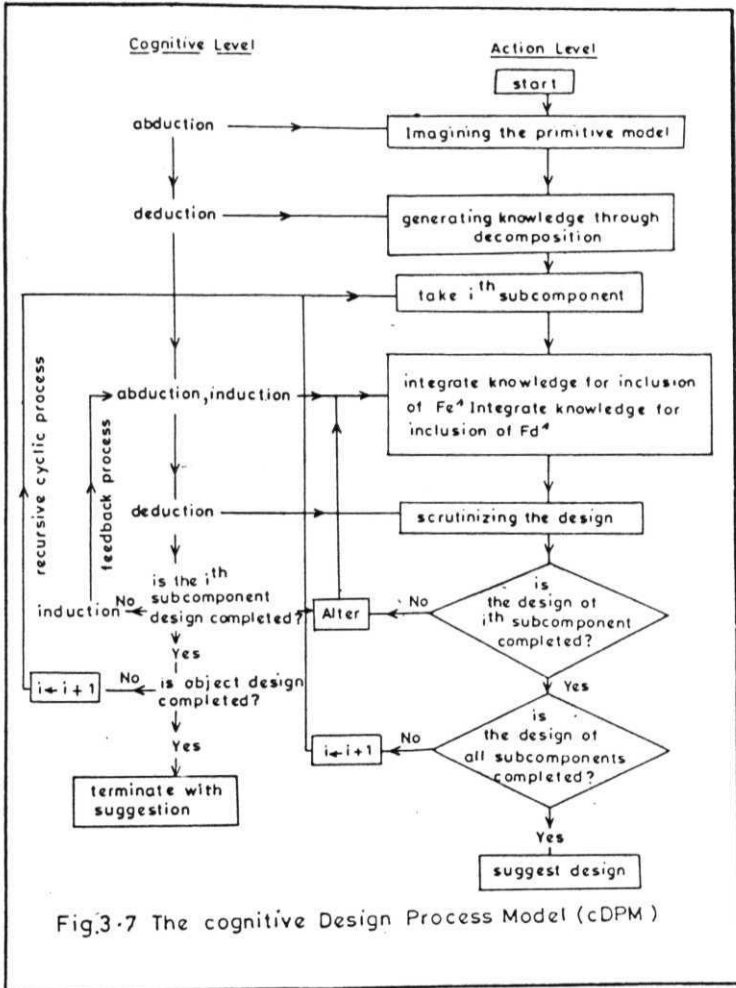


Fig.3-7 The cognitive Design Process Model (CDPM)

first, there exists more or less similar phases in design process and two, to an extent these can be formalized. However, human judgments, perceptions and likings predominate at **certain** locations **of** design which make the design idiosyncratic. **It was observed that** analysis **and** synthesis kinds of practices go interactively in design process. Also are contrasted in this chapter, the earlier models of cognition in design.

Later, cDPM was proposed and identified the reasoning mechanisms such as abduction, induction and deduction at various phases of design. It is concluded that a complete version of intelligent CAD is not possible atleast in the approach under the context of classical cognitivism, but the intelligent assistance through computer is possible. It requires to understand more about what goes on in designer's brain in the context of brain hardware.

## Knowledge Engineering in generic Intelligent CAD System

### 4.1 Introduction

In the first and second chapters, it is discussed, how the design process progresses and various concepts and protocols involved. How human cognitive psychology plays a role in the design activity is described later. In other words, design process is explained both at representative and cognitive levels. In this chapter, an attempt is made to model design activity in systems engineering point of view. Unlike earlier chapters, understanding design at system level is significant in a different way. Here, earlier propositions about design process, planes of **processing** design, knowledge types and their processing will be **implemented** on the system and hence, forms a premise for contrasting theory and practice.

As design is intrinsically<sup>a</sup> complex activity, the role of continuous interaction with human designer is considered essential in I-CAD systems. It was seen earlier, that the decisions and judgments involved are subjective **in nature** and AI techniques are not sufficiently developed to support them. Most of **the** knowledge pertaining to design, though suffers from inconsistency and uncertainty, there is considerable portion of it, whether it is of object, of domain or of design is explicable and can be formalized. In these situations, it is possible for the system to support the designer in an intelligent way.

**Smithers et al(1990)** identified that there are two approaches for studying

design. One way, is to build artificial systems that by replicating, a human designer's behavior, are able to create designs by some means or other. Contrarily, **the** other approach is to try to build systems which provide intelligent **support to** a human designer. They suggested that both ways are AI-based approaches for studying design. Nevertheless, in either case, it is sufficient for the system to show intelligent behavior. **Mac Callum(1990)** also supported this proposition. Adding to this, he suggested that AI programs need not necessarily have to replicate the human behavior. But, the present study views that barring few subjective judgments, the representative and cognitive models of generic activities are analogous to each other and it is possible to intelligently support the designer much akin to as an intelligent human assistant does. In this sense, the restriction of **somehow** which was stated in the Smithers first approach can be eliminated. By finding a sort of unification between the models of cognition and representation, it is possible to emulate human intelligent behavior in the machine not just by **somehow**, rather by integrated understanding. This is what the present study is aimed **at**.

The objective of the present chapter is two fold: One, it discusses **various** aspects of Systems Engineering at every phase of design upon constructing a prototype expert system(for a particular application) and two, it contrasts designs of different domain from Knowledge Engineering point of view, mainly in order to generalize the above prototype expert system features. It is **imperative** that the construction of the expert system is mainly to validate experimentally, the concepts proposed in the last three chapters.

Experimentation in AI is more beneficial, especially, when it is intended to understand the task's nature. Generally, experiments consist of generating a working-system, wherein, the intelligent behavior will be realized. Obviously, in light of this

the selection of problem for experiment is significant. It should reflect properly the conditions of the task nature, eventhough, it appears simple at first glance. Inaddition, it should also contain generality of the task under consideration. An important point to be kept in mind is that the working-system is neither a commercial product nor a prototype rather is an illustrative vehicle. The objectives of the working-system is to demonstrate in a corroborative way, the assumptions and hypotheses underlay in the problem solving strategy of design task.

Design of electrical Power Supplies was considered in this chapter. An intelligent program was built to this effect and observed how it works upon examining the results. This work is further related to the propositions of understanding design.

#### **4.2 Design at System Level and Design Life Cycle Model**

Observation of human designer while he is designing, reveals that there exists, different phases in design process. A primitive model of the design object is conceived initially and is followed by identifying various subcomponents that constitute the primitive model. Here, depending upon the end application of the object, he fixes the values for constraints(constraint satisfaction by parametric selection). In addition, the decisions that are to be taken during various situations of the design are also established. In summary, he knows what he has to do in many cases. On the other hand, sometimes designer may not be clear about its application. In these situations, he can not identify the case suitable to him. Hence, part by part designing starts with intensive interaction with computer. In other words, the system is expected to aid designer's decision making while identifying the subcomponents, selecting the values for constraint satisfaction, choosing

between alternatives. Sometimes, even designer is not **very** confident about the values. In these cases, system should provide information using past experience what has been done earlier.

In systems point of view, there are three different phases identified for design: **the planning, the scheduling and the implementation.**

#### **4.2.1 Design Planning**

With the closed interaction with designer, the system should be able to know what to design, when to design and the sufficiency of knowledge available to design. This part of design can be called , **design planning.** In routine designs,the planning step is embedded in cases / past episodes where component by component hierarchy is **fixed.** Even in less established cases(nearest match) too, there exists a way for planning. However, planning is critical when it is innovative or creative designs. Since, in these cases, the required complete object structure is far from the primitive ones.

#### **4.2.2 Design Scheduling**

Once, it is planned, it is observed that the domain poses a constraint of what to be designed first. This constraint is independent of which function has to be satisfied first. This type of scheduling constraints primarily emerge from the parameters. When, a component's output(in design, the output is always structural parameters) is directly or indirectly required to design another component, then necessarily the first component has to be designed prior to the later. Hence, a sort of **shuffling** of the subcomponents/atom components in component-tree is required which was generated at identification step based on functional satisfaction. This

can be called as **design scheduling**.

#### 4.2.3 Design Implementation

Now, the veritable design starts. Also called the design **implementation** stage, here, the design procedures are followed through suitable substitution of the knowledge of factual and constraint satisfaction in order to produce geometric / structural parameters of the object. A constraint can overcome in a single step or sometimes in a recursive action. Once, the intermediate design parameters are achieved, they have to be analyzed for their validity and if needed have to be redesigned. The effect of these assumed values will in general be subsided during the recursive process. Otherwise, it results in overdesign or underdesign. The combination of design synthesis, analysis of the results and redesigning yield an optimum subcomponent at local level and an agreeable object at global level.

Fig 4.1 shows, how the design progresses at Systems Engineering level and Fig 4.2 compares between system level and representative level descriptions.

It is to be concluded that at system level, understanding design process involves three stages of design as discussed above: planning, scheduling and **implementation**. In other words, it can be termed as **design life cycle model** at system level description.

Contrasting with rDPM, it can be identified that select and decompose phases are implicit in design planning, whereas, integrate, scrutiny and alter phases constitute design scheduling and implementation. An important point which is worth discussing is about innovative design. In DPT innovative design is postulated as the fulfillment of desired functions **in** addition to essential functions upon amalgamating the knowledge of essential aspects with inducted knowledge of desired

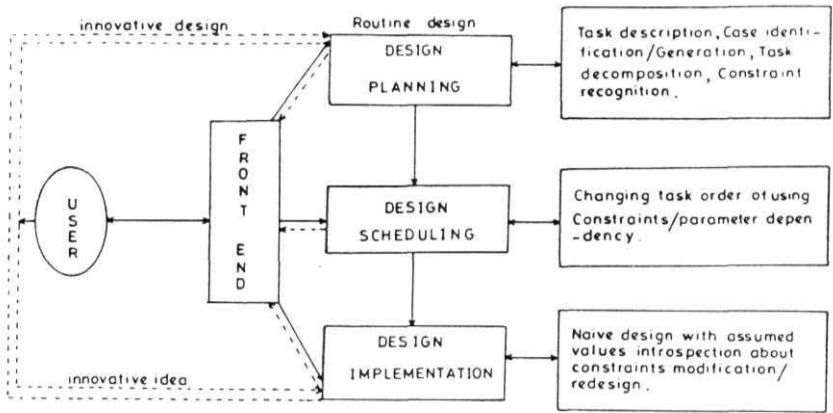
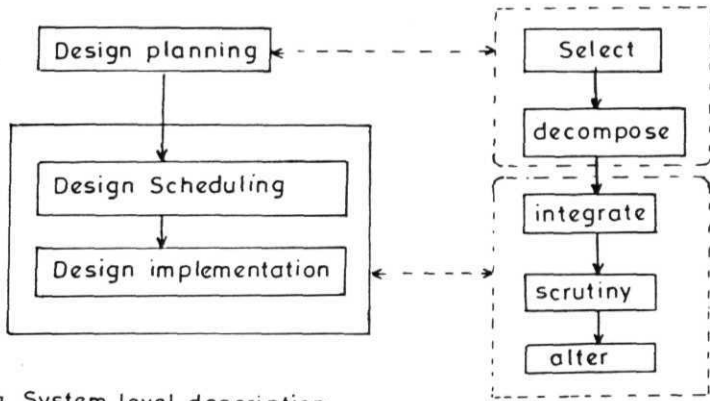


Fig. 4.1 Modelling design process at system level. Design Life Cycle Model



a. System level description

b. representative level description

Fig.4.2 Comparison between System and representative level description

aspects. At systems level, this can be done at two stages: at planning stage, the identification of subcomponents by functional satisfaction triggers innovating idea and/or at implementation stage, the assumptions and trades off to overcome the constraints trigger. At system level, when the idea comes at later stage it needs to replan, schedule and implement.

In the following sections, an experimental working-systems' design is presented taking a case of Power Supplies. Two more examples, one from chemical engineering(Thermal design of Heat Exchanger) and one from medianical engineering(Mechanical design of Screw Jack) are considered. The aim of studying these two examples are two fold: one, it is tried to establish that design process is a generic activity and two, it is <sup>to</sup>prove that design follows more or less **similar** process in problem solving techniques at systems level. The case of developing an expert system for Power Supplies design is identified, as a good illustrative vehicle, which is sufficiently general, to explain the propositions and understanding of generic design activity, as the present study proposes.

### **4.3 Analysis of Design Process in the Domain of DC Power Supplies**

The problem presented here is relatively straight forward. The task is to design a DC Power Supply(Ramamoorthy, 1987; Raina et al, 1991; Nessler and Stadelmeier, 1991; Sasikumar, 1992; Shah, 1993), which ultimately be used, in most of the electronic circuits, which need constant DC Power Supply, ranging from microprocessors to electronic toys. A typical design problem of DC Power Supply will appear as:

**Design a DC Power Supply to produce DC output of 100 VA rating to a 15 fl resistance load having a ripple factor better than 0.05 given**

**line voltage of 230 Volts at frequency 50 Hz(Indian conditions).**

One can observe from the above task description that the designer under certain end requirements, i.e. **for** satisfaction of functions(defined **in** the parametric plane) has to evolve a DC supply with all structural parameters. Let for generality, user/designer is free to take any decision on parameter value(s) either based on his own experience or based on end application of object under design or based on the advice of system or transfer the decision making to the system as a default.

The first step in designing is the selection of a primitive form of a DC Power Supply. There may be many cases in the past **experience**, which are nearly or truly matching to **what** is required by the task. Hence, a case is selected. This primitive form is synthesized in the following steps.

**Fig. 4.3** and **Fig. 4.4** represent block diagram and circuit of the DC Power Supply respectively. Here, for stepping down the voltage, a **transformer** is used; for converting AC to unidirectional wave form, a rectifier is used and for reducing the ripples thus providing continuous and constant DC supply, a filter is used. It can be observed that from the diagram for every function, there corresponds a structural component.

#### **4.3.1 Design of the Transformer**

Before attempting transformer design, it is obvious to know which kind of transformer has to be designed. There are many standard varieties available such as small, power and distribution transformers. Again, they are **of** two types, shell type and core type. Knowledge in the form of constraints satisfaction is available to select by the user the kind of transformer required for the purpose under consideration. Otherwise, the end application of the DC supply and/or

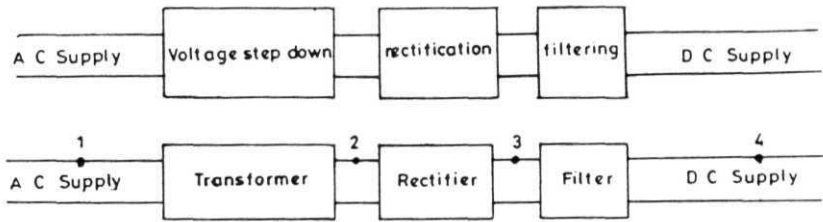


Fig.4.3 Block diagram of DC power supply function vs component

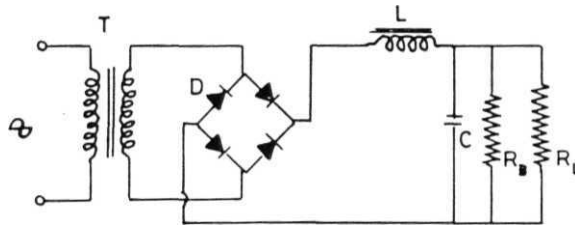


Fig.4.4. DC Power supply

- |                        |                     |
|------------------------|---------------------|
| T Transformer          | D Diode             |
| L Inductance           | C Capacitor         |
| $R_B$ Bleeder Resistor | $R_L$ Load Resistor |

description of the problem automatically fixes the choice of transformer. As an illustration, for stepping down from 230 V to 15 V a small power transformer of shell type is recommended because the constraints such as lower power rating and minimizes-leakage-reactance have to be observed.

The same procedure is repeated for the selection of every subcomponent. The application, the constraints, the user either individually or by interacting with the system ascertain the type of subcomponent to be used. This way of selection always precedes every designing step of subcomponent. So, in **the** following discussions, whenever a design procedure is described for a given **subcomponent**, it is imperative that the subcomponent is selected undergoing the above mentioned methodology. **Appendix- A** briefs the functions and the constraints that have to be satisfied by every element of component-tree.

- (i) **Design of the Core**

$$\phi_m = \frac{1}{4.44 \times f \times T_e}$$

$\phi_m$  is the maximum flux in the core,

f is AC supply frequency and is given in the problem

$T_e$  is turns per Volt; read from Table 4.1 (VA Vs  $T_e$ )

VA is Volt Ampere rating

$$A_i = \frac{\phi_m}{B_m}$$

$A_i$  is net area of the core

$B_m$  is the maximum flux density, assumed by user / constant  
for given application

Gross core area,

$$A_G = \frac{A_i}{stf}$$

where, stf is stacking factor and its value is assumed as 0.9

Square cross sectional core minimizes the total length  $l_w$  of the wire **required** for windings. Hence square cross sectional type is selected.

The core can be made up of E-I or T-U or E-E stampings. The user/system(default) can decide the type of stamping and the corresponding parameter values are read from respective Tables. The E-I stamping was considered as a default for the present example.

Width of Central limb

$$A = \sqrt{A_G}$$

Read from **Table 4.2** for given type of stamping and A, the values of D, C, D, E and sankey number(see Fig. 4.5).

If A is not a standard value, the nearest next highest value of A is taken and other parameters are read.

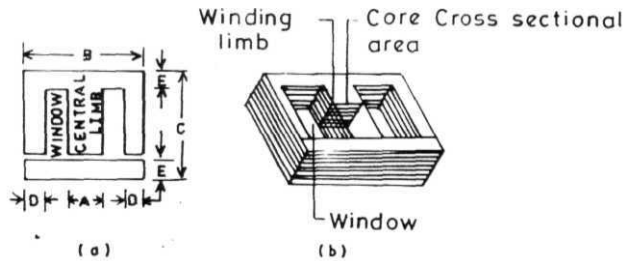


Fig.4-5

(a) Shows an E-I stamping

(b) Shows a number of E-I stampings assembled to form the core of a transformer

$$\text{Window area} = \left( \frac{B - 2D - A}{2} \right) (C - 2E)$$

• (ii) Design of the Windings

Select the appropriate winding material (APPENDIX- A) .

— (ii) a) The design of the Primary Winding

input primary current,

$$I_p = \frac{\text{output VA}}{\eta V_p}$$

output VA is Volt Ampere of the transformer to be designed,

$\eta$  is efficiency of the transformer and

$V_p$  is transformer primary voltage

$$a_{p1} = \frac{I_p}{\rho_c}$$

$\rho_c$  is current density of primary winding material

diameter of the primary winding,

$$d_{p1} = \sqrt{(4 \times a_{p1} / \pi)}$$

$d_{p1}$  so calculated may not be an available standard diameter. Hence from **Table 4.3** (IS: 1595 - 1960), nearest standard diameter has to be chosen. This table, contains for a given insulation, the diameter and thickness class of wire.

Now, from this standard diameter, the cross-sectional area,  $a_{p1}$  can be calculated.

Number of turns,

$$T_p = V_p \times T_e$$

**(ii) b) Design of the Secondary Winding**

current in the secondary winding,

$$I_s = \frac{VA}{V_s}$$

$V_s$  is secondary winding Voltage of the transformer  
cross-sectional area of secondary winding conductor,

$$a_{st} = I_s / \rho_c$$

$$d_{st} = \sqrt{\frac{4 \times a_{st}}{\pi}}$$

The procedure of designing follows the same course as that of **primary** windings.

• **(iii) Procedure for Checking**

It is required to check whether the designed object structural parameters are correct. Hence, the design undergoes the scrutiny phase. Here, the engineering analysis procedure is to find window area available is sufficient to window area required.

$$\text{Window area available} = \left( \frac{B - 2D - A}{2} \right) (C - 2E)$$

from core design

$$\text{Window area required} = 1.25 \left( \frac{T_p \times a_p}{S_{fp}} + \frac{T_s \times a_s}{S_{fs}} \right)$$

from windings design

$S_{fp}$ ,  $S_{fs}$  are space factors for primary and secondary windings respectively.

If window area available is greater than window area required **then** design is said to be satisfactory. Otherwise, the stamping of higher size **is** taken and core is altered. This recursive design (scrutiny - alter) terminates when the above condition satisfies.

#### 4.3.2 Design of the Filter

A filter is used in the Power Supplies mainly, to reduce the **ripple** and continuously supply the DC. There are many varieties of standard filters available, among which inductor filter, capacitor filter, LC filter, CLC filter and CRC filter are usual. Among these, one is selected considering among others the ripple factor. However, the complete selection criteria is presented in **Appendix- A**.

Let us consider the selected filter type is LC filter.

Voltampere (VA)	Turns per volt( $T_e$ )
10	23.3
15	17.5
20	14.0
25	11.7
50	7.0
75	5.6
100	4.6
150	4.0

Table 4.1 Number of Turns per Volt,  $T_e$  Corresponding to Voltampere Rating VA.

Sankey Type Number	A(mm)	B(mm)	C(mm)	D(mm)	E(mm)	Remarks
17	12.7	38	31.8	6.35	6.35	
30	20	60	50	10	10	4 holes 4mm dia
31	22.2	66.6	55.6	11.1	11.1	4 holes 4mm dia
45	22.2	66.6	55.6	11.1	11.1	4 holes 4mm dia
15	25.4	76.2	63.5	12.7	12.7	4 holes 5.55mm dia
44	25.4	76.2	63.5	12.7	12.7	4 holes 5.55mm dia
14	25.4	84.0	66.6	13.5	13.5	4 holes 5.55mm dia

Table 4.2 Dimensions of E-I Stampings

Nominal conductor diameter(mm)	Fine covering (mm)	Medium covering(mm)	Thick covering(mm)	Extra thick covering(mm)
0.250	0.283	0.301	0.319	0.344
0.258	0.291	0.309	0.327	0.352
0.265	0.299	0.317	0.335	0.362
0.280	0.316	0.334	0.351	0.379
0.300	0.336	0.354	0.372	0.400
0.307	0.345	0.362	0.380	0.408
0.315	0.354	0.372	0.389	0.417
0.335	0.375	0.393	0.410	0.441
0.355	0.397	0.415	0.432	0.465
0.375	0.418	0.438	0.455	0.489
0.400	0.445	0.465	0.488	0.518
0.425	0.472	0.493	0.513	0.549

Table 4.3: IS: 1595-1960 Round Copper Wire(Synthetic Enamel)

Inductance (Henry)	Current rating (ampere)	resistance ( $\Omega$ )
0.1	1	1
10	0.1	200

Table 4.4 Inductor Values

ripple factor,

$$r = \frac{\sqrt{2}}{3} \times \frac{1}{(4\pi f)^2 \times L \times C}$$

L is the inductance and

C is the capacitance

$$L_{critical} \geq \frac{R_L}{3 \times (2\pi f)}$$

Find from **Table 4.4** equal or the near highest standard L value, by giving  $L_{critical}$  and current rating of the load.

From ripple factor equation,  $C_{estimated}$  value can be evaluated, since ripple factor desired is provided either by user or by the kind of application and L value is known. Find, from **Table 4.5**, the equal or the near highest standard C value by giving  $C_{estimated}$  and Voltage rating of capacitor values.

The ripple factor with the modified inductor and capacitor values can be calculated by again revoking the ripple factor equation.

Select the Bleeder Resistance ( $R_B$ ) from **Table 4.6** by giving the estimated value of Bleeder Resistance ( $R_{B_{estimated}}$ ) and power rating of bleed resistance.

$$R_{B_{estimated}} \leq 3 \times (2 \times \pi \times f) \times L$$

Capacitance	voltage rating (Volts)
4	450
8	400
8	450
10	400

Table 4.5 Capacitor Values

resistance ( $\Omega$ )	Power rating (Watts)
91	3
91	5
100	3

Table 4.6 Bleeder Resistor( $R_B$ ) Values

Type	PIV	Forward Current	Forward Voltage	Forward Resistance
ln4001	50	1	1.1	1.1
ln4002	100	1	1.1	1.1
by397	200	2	1.1	0.55
by399	800	1	1.1	0.55

Table 4.7 Diode Values

power rating of  $R$

$$= V_{DC}^2 / R_B$$

### 4.3.3 Design of the Rectifier

A rectifier is required to convert AC voltage(bidirectional) to unidirectional voltage. There are different types of rectifiers are available e.g. half wave, full wave and bridge rectifiers. However, the basic component is diode in them.

To select a diode, the maximum secondary voltage of **transformer i.e.**, Peak Inverse Voltage(PIV) and Peak Current are required, which are the outputs of transformer secondary winding, the diode(characterized by diode number) is selected. Bridge rectifier was selected, for the given problem (**Appendix - A**).

Voltage drop in LC filter,

$$V_F = R_{inductor} \times I_L$$

$R_{inductor}$  is resistance of the selected inductor

$I_L$  is Load current

Net DC voltage required =  $V_{DC} + V_F$

$$V_P = n \times \frac{\pi}{2} (V_{DC} + V_F)$$

correction factor,  $n = 1.2$  or 20% high

$$V_{rms} = \frac{V_P}{\sqrt{2}}$$

$V_{rms}$  is Root Mean Square Voltage of the secondary windings of transformer

The diode is selected by looking into the **Table 4.7** by matching the equal or near highest PIV. In case more than one type of diode satisfies this requirement, the next criteria is considered the current rating matching. Further, the forward resistance will be the criteria, when more than one diode satisfies both PIV and current rating, diode with lower forward resistance value is preferred to higher values.

Upon observing the design of the three components leading to an integrated object(DC Power Supply) the following can be understood:

1. The components, transformer, rectifier and filter in fact are individual components having their own design pattern. However, when they are integrated, few relations which are in the form of

common constraints satisfaction, bring them together to synthesize. The constraints take the form of structural compatibility and/or commonality in performance qualifiers etc. The knowledge pertaining to these relations is called relational knowledge and is essentially of domain in nature.

2. Though, the order in the functional satisfaction generates a task tree(component design is the task), this order need to be shuffled due to the parametric dependencies. This leads to another phase of design called scheduling and is intermediate to design planning and implementation.

In the example of Power Supplies design, stepping down of the voltage is caused by transformer. However, due to the inherent voltage drops in rectifier and filter components, the transformer has to be overdesigned to this extent, above the DC voltage required by the problem definition(load).

Voltage of Secondary winding,

$$V_P = V_{DC} + V_R + V_F$$

$V_R$  is Voltage drop in rectifier

But unless exact **PIV** and peak current from transformer's secondary windings are known, the diode of rectifier cannot be designed. Let this be constraint 2 for scheduling. Based on the constraints 1 and 2, the schedule can be prepared. The first component to be designed is the filter. Among transformer and rectifier, transformer has to be designed earlier to rectifier, since, the

comparison between voltage drop constraint and diode selection, though transformer output constraint suggests that the later is more significant than former.

3. The scrutiny and alter is generally based on engineering analysis, wherein, many times, the structural compatibility and performance qualifiers' accordance decide the adequacy of the design.

#### **4.4 Expert System Development**

The essential components of any expert system are the Knowledge Base and Inference Engine (Rolston, 1988; Bahrami, 1992; Frost, 1986). Because of the degree of separation between these two components, the Conventional programs and expert system programs differ. The various kinds of knowledge, **principally**, the procedures and facts are stored in the Knowledge Base. The Inference Engine locates Knowledge in addition to infer new Knowledge from the base Knowledge (Fig. 4.6 shows a part of expert system architecture).

##### **4.4.1 Expert System in Design**

As was recognized earlier, design activity demands a good user interaction. Here, user has to continuously provide the functions and orders of priority of satisfying these functions, judge the **efficacy** of the evaluated values etc. to mention a few. Similarly, design activity necessitates, at various stages of design, various predominant inference mechanisms. This makes Inference Engine design not to use any single inference mechanism. Rather, it will be driven by the paradigm. Similarly, various kinds of knowledge possess various characteristics. Hence, it is required to represent them according to their characteristics keeping in mind the

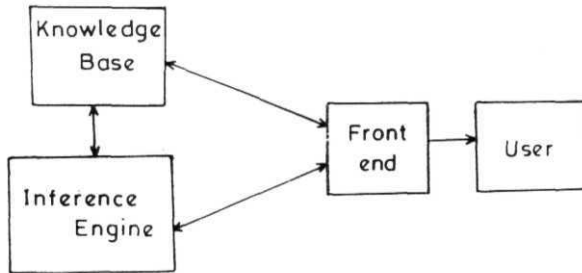


Fig.4. 6 E S architecture

flexibility, accessibility, maintenance, updation and transparency offered by the representing type. Often, more than one kind of representation is required.

#### 4.4.2 An Experimental Working-System Development for DC Power Supplies Design

'C' language is selected for implementation of the Program, which offers sufficient flexibility to the programmer while designing, integrating and implementing the program modules. The Knowledge of object, of domain and of design are organized as was discussed below. The Inference Engine necessarily follows the paradigm proposed in the rDPM. The overall system organization follows the model at system level description.

Since, the working-system is paradigm driven, a control software controls the

various phases of design activity depending on the context. The control software is menu driven.

All the titles of the cases / applications which are already designed are **displayed** such that the user identifies one of the application that matches to his application / case. As a default, a general purpose DC Supply can be chosen by the user. This mechanism is abduction. The rule in the Knowledge Base looks like,

IF           (application is receiver)  
THEN       (Power Supply is DC1)

In case the application is not known, the control probes the user about the functional needs of the object that are to be satisfied. Once the user provides the functions then the control matches the nearest type of DC Power Supply or default design.

For instance, DC1(a specific type of DC Power Supply) can be selected based on a few constraints. If application is known, it automatically infers that these constraints are satisfied. Contrarily, if the user is not clear of application but knows the functions then too DC1 can be selected.

A typical case is represented as in frame 4.1

This primitive structure is a basis for getting further sub-structures in decomposition phase.

The next phase is decomposition. Since, every component including object has to fulfil certain functions, functional satisfaction is the key to identify **the** general

<b>object</b>	<b>DC1</b>
<b>function 1</b>	<b>stepdown</b> in AC voltage
<b>constraint 1</b>	small power rating
<b>constraint 2</b>	leakage reactance is to be minimum
...	
<b>constraint n</b>	
<b>function 2</b>	rectification
<b>constraint 1</b>	higher efficiency
<b>constraint 2</b>	more conduction
<b>constraint 3</b>	less ripple
<b>constraint 4</b>	less Peak Inverse Voltage
...	
<b>constraint n</b>	
<b>function 3</b>	filtering
<b>Constraint 1</b>	more load current
<b>Constraint 2</b>	more regulation
<b>constraint 3</b>	ripple is independent of load
...	
<b>constraint n</b>	
<b>domain knowledge</b>	concerned domain facts & procedures
<b>design knowledge</b>	concerned procedures

4.1 A typical **Frame**

form of the component at any level. There is a reason why the word general is used. Consider an example.

IF (voltage is to step down)  
THEN (general form of component is transformer)

IF ((general form of component is transformer)  
and (power rating is low)  
and (leakage reactance is minimum) )  
THEN (specific form of component is SSPT)

SSPT is small power shell type transformer

Once the type of specific form of the component is known, then knowledge of domain and design are identified and brought to the system, because **the knowledge** is well established in the universe for this type of component. The process of identification and choosing among alternatives by decomposition continues till atom components are arrived (one can see that as a consequence of decomposition, a **component-tree** is formed). This type of inferencing is deduction.

After identifying the subcomponent structure against functional fulfillment, the next job is to attach this substructure to the parent component structure as in a tree. The construction of tree procedure will be continued until all subcomponents / atom components are arrived. This tree represents exactly the case. A case for the considered Power Supply contains knowledge of functions, constraints,

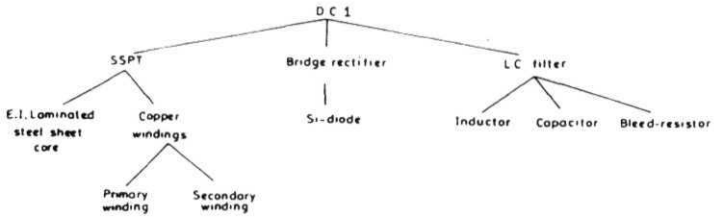


Fig.4.7 Tree after decomposition phase

structures(object knowledge), domain knowledge and design knowledge. Fig. 4.7 shows the tree structure obtained after decomposition step.

Then synthesis phase is enabled with four sub-phases viz. scheduling, integrate, scrutiny and alter. In scheduling phase, the subcomponents/ atom components are shuffled according to the priorities. The shuffling can be performed either with user interaction or as default option. This knowledge which is heuristic is used to a certain extent to reduce the complexity of the design activity. The shuffled tree looks like as in Fig. 4.8

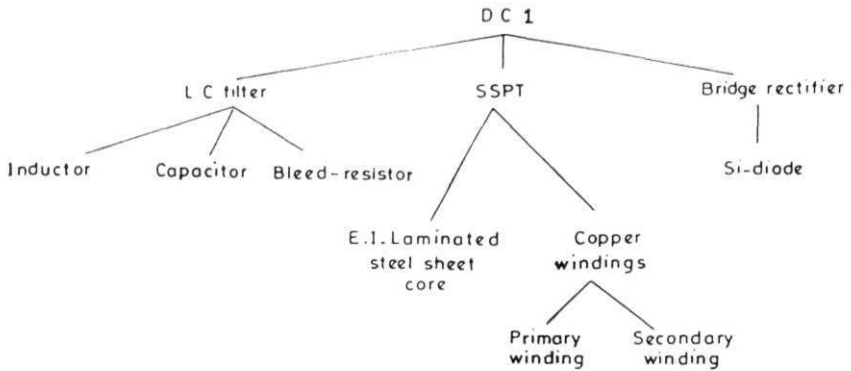


Fig. 4.8 Tree after shuffling

A typical rule for shuffling is as

IF (component is DC1)  
 THEN ( (priority of SSPT is 2),  
 (priority of LC filter is 1),  
 (priority of bridge rectifier is 3) )

IF (component is SSPT)  
 THEN ( (priority of core is 1),  
 (priority of windings is 2) )

Design of any subcomponent is resulted in three subphases: integrate, scrutiny

and alter. An atom component (leaf in component-tree) is chosen either from the tables or expected from the user to provide the necessary specifications / attributes. Travel in the tree is envisaged as recursive postorder depth-first traverse.

At any stage of tree traversal, design of a subcomponent is under progress first in knowledge plane and later in parametric plane. The scrutiny and alter of designed component essentially follows the same way as described earlier. The rules of scrutiny looks like e.g.

```
IF      ( ( expected ripple - obtained ripple ) < tolerance)
THEN   (designed-filter is accepted)
ELSE   (designed-filter is rejected)
```

```
IF      (designed-filter is rejected)
THEN   (enable alter-phase)
```

Recall that the objective always focused at the optimal final object design but not necessarily on reaching the local subcomponent's ideal **performance**. This constraint is overcome by using dependency backtracking algorithm. Here, dependency suggests that the redesign will be undertaken for those sub-structures, which were projected to alter phase. This process terminates, when the expected behavior of the resultant object (here Power Supply) nearly approaches the desired behavior.

## 4.5 Further Design Examples

### 4.5.1 Screw Jack, an example of mechanical engineering Design

The general purpose of a Screw Jack is to lift the load. An every day popular application of it, may be the lifting of automobiles for repairs. There may be many machines to lift the loads such as lifts, cranes, hoists, hydraulic jacks etc, to mention a few and Screw Jack is one such **machine**. The principle behind Screw Jack is explained as follows.

Upon rotating the screw, it exerts two types of motions: One, the pure rotation and another, the linear motion in the direction arranged/such a way as to lift the load. The process of design is to prevent the pure rotation and use the **translational** component of motion to cause the load to move upwards, since translational component only can cause the load to move upwards.

For the sake of various advantages such as portability, lower human effort, rugged conditions of the ground and simplicity in construction, the principle of screw is extended to the form of Screw Jack. As can be seen from Fig. 4.9 it can be observed that, it contains mainly a lifting mechanism and a supporting mechanism (base). The component tree / task tree of Screw Jack system is shown in Fig 4.10 after shuffling.

Lift causing mechanism further contains a nut and Screw, a loading platform, handle and its connecting head and stopper end washer & screw. The supporting mechanism or base contains a base structure and a grub screw. The functions and constraints to be satisfied for every subcomponent are given in **Appendix- B**. The design procedures for every subcomponent are also listed in the same Appendix. For a complete discussion of codes, procedures, tables and designing methods see **Hall, 1980; Maitra et al, 1985; Metin & Rezai, 1986; Pandya & Shah, 1987.**



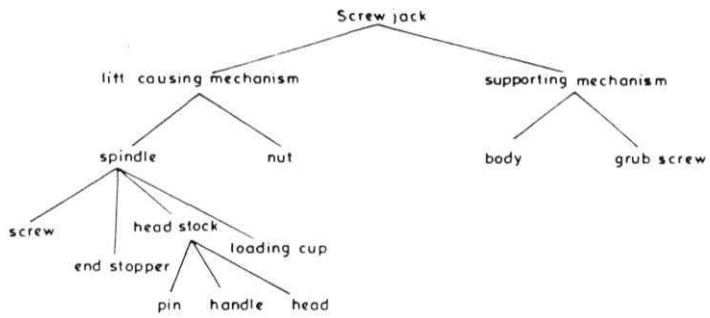


Fig.4-10 Component tree of screwjack system(after shuffling)

As a structure, the spindle is connected to four other **subcomponents** at four different locations. Hence, the design of spindle influences four subcomponents. This adds complexity to the designer in deciding what to do next. This type of **problem**, where structural interaction to many subcomponents at a time, is a common phenomena in mechanical engineering. In fact, one of the reasons for selecting this problem is to show that the rDPM circumscribes these type of problem too.

From the analysis of functioning of working expert system on Power Supplies, it can be identified that decomposition based on functional satisfaction also aids in reducing complexity. Identifying the smaller and smaller components, which have lesser constraint dependencies on the adjoint subcomponents to satisfy eliminates the problem of solving all constraints at a time. In other words, the **complexity** reduces. Eventhough, it is difficult for a mechanical engineer to visualize a screw separate from that of nut, as far as design is concerned, such a visualization usually reduces complexity when screw is linked to several components including nut.

#### **4.5.2 Thermal Design of 1,2 Shell & Tube Heat Exchanger**

In the previous chapters, the issues relating to Heat Exchangers were discussed in the context of design. Recapitulating, a Heat Exchanger is said to be an equipment which transfers thermal energy (heat) from one fluid to another fluid. Let the present argument be restricted to liquid-liquid heat transfer (liquids and gases are kinds of fluids). In chapter 3, a double pipe Heat Exchanger was described which is a class of Heat Exchanger, Since, double pipe Heat Exchanger is too simple example for design to describe, a shell & tube Heat Exchanger design is

considered which offers sufficient material to analyze.

In a subjective sense, design is a process of realization of some idea / concept(s). In this sense, it is not always necessary for any component to possess geometric / structural attributes that quantify its description. For example, in chemical formula design, the components are the chemicals and the analogical structural attributes of design are for e.g. purity, melting / boiling point, viscosity, refractive index etc. depending on the material and the weight percentage of that material in the overall formula. Similarly, shell & tube Heat Exchanger design, in view of thermal energy transfer is something different from mechanical engineering point of view. In the later case, the emphasis is on in addition to best possible thermal energy transfer, the mechanical forces like internal pressures, hoop stresses, buckling moments, mechanical failure occurrences etc. The present argument considers the thermal design only aiming at efficient thermal energy transfer.

Now, it is easier to realize why thermal design of 1,2 shell & tube Heat Exchanger design is selected as a case study. The point here is that, the present study primarily concerns about the generic nature of design, whether it always leads to physical / structural attribute quantifications or not. It argues that design is of more ontological activity. Another point considered is the nature of that task. The thermal design of Heat Exchanger can be categorized as a task of inconsistency at design procedures. Many of the parameters which are either the design process intermediates or final designed components' description values(for example, outer diameter of the tubes, inside diameter of shell, tube-tube distance etc.) come directly from the broad assumptions. Many times, there is no way to test whether they are correct / near correct at the first place. Based on these values

prima facie, the later part of the design is completed. Only then there will be a chance to realize whether the assumed values are true. This type of inconsistency in parametric values lay stress on scrutiny and alter procedures.

**Appendix- C** shows, how the task is to be worked out in shell & tube design and in addition to the stepwise procedure of the design. A typical configuration of this Heat Exchanger is presented in Fig 4.11. Component / task tree is shown in Fig. 4.12. For a complete description of design procedures see **Kern, 1950; Perry & Chilton, 1983; Fanaritis & Bevevino, 1979; Lord et al, 1979; Yokell, 1990; Guyer et al, 1989.**

While observing the design, one might have observed that the description of subcomponent(not necessarily structural) is fixed with less effort. For example, the length, diameter, gauge(thickness), material of construction which are the description of a tube in view of thermal design are all assumed. A weak qualitative knowledge / past experience is available to guide the user for this purpose. However, when entire design is completed, there are rigorous mathematical procedures available to check the validity of the assumption. The heat transfer coefficient, which is indicative of good thermal energy transfer and the net pressure drop, which is a qualifier of pumping energy loss due to friction estimated are used to qualify the assumptions made at different design steps. Later, a judgment is made to select such an object-description-values that correspond to highest possible heat transfer coefficient value and lowest possible pressure drop value that occur simultaneously. The inconsistency caused is eliminated only after **complete** design at the scrutiny and alter phases. This is the reason why few researches called design as a process of redesign.

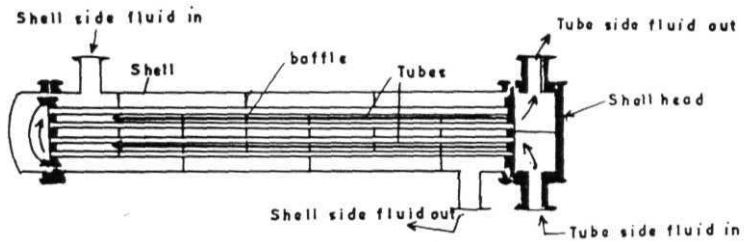


Fig.4.11. 12 Shell and tube heat exchanger

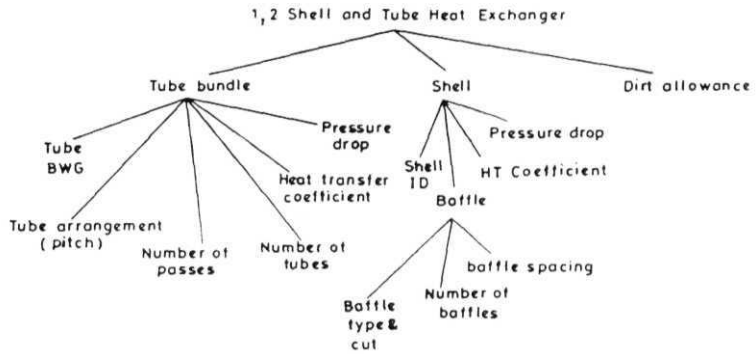


Fig. 4-12 Component tree/Task tree of 1,2 shell & tube Heat Exchanger

#### **4.6 Analogy between designs of DC Power Supplies, Screw Jack and 1,2 Shell and Tube Heat Exchangers**

Contrasting examples from three Engineering faculties, electrical, **mechanical** and chemical, the following points are observed :

1. Design is an evolutionary synthesis process whether it is the case of DC Power Supplies or Screw Jacks or Heat Exchangers. A finished object structure is synthesized by evolved individual subcomponents.
2. Imagination of primitive structure of the object is the kernel to design process to proceed to any extent. This can be seen in these three examples.
3. In case of DC Power Supplies, for every function there exist a subcomponent. Nevertheless, Screw Jack is a case where many-to-many relations exist between functions and components. In case of Heat Exchanger design, the identification of functions and components is rather ambiguous. However, it can be said that for every logical entity, there are many-to-many relations exists between corresponding logical functions. Upon summing, it can be suggested that the artifact to be designed is composed of subcomponents. They are distinguished by their independency but are associated with others through the common functions to be fulfilled. Hence, in DC Power Supplies the independency is very clearly seen. This is mainly due to one-to-one correspondence existed between components and functions. As this correspondence changes to one-to-many or many-to-many, the components try to associate more and more and thus complexity increases and complexity-induced problems enhance.
4. In case of thermal design, many of the attribute values are assumed initially.

There is no way to check whether the assumed values are true or not. This poses inconsistency in problem definition. This type of inconsistency is also seen in Screw Jack design but not to this extent. In case of DC Power Supply the inconsistency is almost not prevailing. This is reasoned partially to the clarity in domain knowledge and low component dependency that can be seen while observing the expert system. In case of DC Power Supplies, all phases are equally significant, whereas in the case of Screw Jack, the emphasis is on integrate phase.

5. In every design of object, the particular subcomponent to be designed is selected among alternatives based on constraint satisfaction.
- C. Knowledge of object description attributes need not necessarily be the structural parametric values.
7. Scheduling which subcomponent has to be designed first always precedes the design synthesis and is an almost essential job.
8. Though, the subcomponent exists individually with its own boundaries, it is linked either objectively i.e. by structural compatibility or subjectively. i.e. For example, the total energy given from tubes(by hot liquid) is equal to the total energy received by shell(by cold liquid) in case of thermal design or sum of the voltage drop due to filter & rectifier and voltage requirement by load is equal to the voltage delivered by the secondary windings of the transformer in case of Power Supply design are subjective relations.
9. Any design problem though apparently looks simple, it possess all the characteristics of generic design process. The tasks differ in degree rather than in contents. This observation reiterates the generic nature of design activity.

10. Design procedures produce the components which will have essential characteristics. However, the improvements are possible within the constraint of availability of knowledge. The clues for improvements come from the subcomponents which have additional constraints that can be fulfilled in addition to that of essentials. This type of improvements are different from innovative design by definition.
11. Comprehensively, design as a process is not limited to epistemological level of human being rather is ontological involving social and technological developments.

#### 4.7 Summary

The present chapter discusses various aspects of Systems Engineering at every phase of design upon constructing a prototype expert system for a particular application. In addition it contrasts different domain designs from Knowledge Engineering point of view, in order to generalize the above prototype expert system features, to validate experimentally, the concepts proposed in the last two chapters. At this juncture, how design progresses at Systems Engineering level i.e., design life cycle model is suggested and design process is compared between system level description and representative level description. It is proposed that design process involves three stages of design : planning, scheduling and implementation, which can be termed as design life cycle model at system level description.

In summary,an attempt was made to understand the designs of various domains taking three examples; a DC Power Supply from electrical engineering, Screw Jack from mechanical engineering and 1,2 shell & tube Heat Exchanger from chemical engineering./ DC Power Supply example which represents a general

case is taken to explain the design characteristics by realizing an working expert system. Screw Jack is a case of complexity intensive design, whereas Heat Exchanger design is inconsistency dominated design. However, it is shown that the characteristics of working-system for DC Power Supply, are suitable for these cases too. As a consequence, it can be concluded that the working-system posses all the characteristics of generic design as explained by rDPM and cDPM.

## Conclusions and Future Directions

*The Literature* shows that the selected topic on design tasks is relatively recent in nature. Even the basic questions about design task structure in the perspective of AI are not addressed thoroughly and the work so far published reveals that this area is yet to be understood from its roots. Hence, the study is primarily aimed to apprehend design tasks from representative and cognitive angles of view.

Design is well known as a creative human activity and is a process of providing form from a formless description. It was identified in the present study that there are four reasons why design tasks are difficult to automate.

1. *The judgments involved are subjective in nature.*
2. *Grasping relation in associations among various components is contextual and human specific.*
3. *Design needs developed techniques of pattern matching from past experience and*
4. *Computers are best for deductive logic whereas design demands larger proportions of abductive and inductive reasoning mechanisms.*

While keeping all these limitations in view, the present study starts with answering such questions as

1. *Design: What it is in the perspective of AI ?*
2. *What is the morphology of design tasks at the representative level of understanding design?*
3. *What is the epistemological model of the human design expert while in active state ? and*
4. *How design process is understood and modeled at system level, after conceiving the concepts arised while answering the above questions?*

## 5.1 Summary **and** Conclusions

The validity of design as a task for AI community was discussed in the introduction. It was recognized that the understanding design tasks are significant in light of their complexity, inconsistency and intuitive behaviors involved. Besides understanding what is and how design progresses in view of researchers, the present study makes a critical contrast among various models that were proposed earlier. It was contemplated here, that this critique leads to constructing a global model for design activity. Both at representative and epistemological levels, the critique was made, principally to make a philosophical insight of design. The earlier work in this regard is categorized broadly in two ways: exploratory and search. Exploration based models consider the primitive description of design object as a starting point for any design to be initiated and try to refine this structure. On the other hand, isolation of desired strategy in the solution space is a principal technique in search based models. The model proposed in this study, is inclined towards the former kind of model . This conclusion was arrived at upon identifying and defining the design task as an evolutionary synthesis process.

Chapters 2 and 3 are intended primarily to model the design tasks at representative and epistemological levels respectively. Our understanding of design begins with classifying design task as creative, innovative and routine, which ofcourse are first coined by Gero(1990). Here, it is argued that design activity is principally a synthesis task as the knowledge out of primitive structure is accrued with various concepts, components and knowledge at every step of design process in order to get the final structure of the object.

Comprehensively, the understanding of the design activity is put as a goal oriented, evolutionary synthesis process demanding judgment and decision support. The objective of the second chapter is two fold. It describes **rDPM, the representative Design Process Model** and secondly, provides a conceptual base to rDPM by proposing a **Design Process Theory (DPT)**. Chunking of primitive structure, establishing linkages(relations) with other subcomponents and integrating or assembling them to achieve the final object, constitute the subject of rDPM. The present study, besides accepting earlier ideology on design, views it rather differently at the basic level itself. It is argued that a further classification of function set into two sub-sets viz essential function set and desired function set makes the understanding of design task more transparent and embodies different models proposed earlier. The idea of splitting up of function set explains Gero's classification of design task with higher clarity. Making the ill-defined(innovative and creative) design task more explicit is an implicit objective of this study.

The proposed Design Process Theory suggests that design works out at two planes: knowledge plane and parametric plane. Also, it<sub>1</sub> recognized that there exists, for a given subcomponent, one-to-many relations between the elements of these two planes i.e. each term in<sub>1</sub>knowledge level description is related to a

corresponding term in parametric level description.

Proposition 1 defines design as a process of satisfying  $F^e$ (set of essential functions) completely and to an extent  $F^d$ (set of desired functions) depends on the ability and context, whereas, proposition 2 links  $F^e$  and  $F^d$  at both knowledge plane and parametric plane and identifies that there may be some common parameters exists in common to  $F^e$  and  $F^d$ . Here, the origins where  $F^e$  and  $F^d$  are stemmed from is identified. Knowledge of functional aspects of the object including the structure, operations and their constraints are generated during analysis of primitive structure and this is precisely what was named as essential aspects of design or in other words elements of set  $F^e$ . Elements of  $F^d$  stems from the designer's thrust to overcome as many constraints as possible while synthesizing the object. This is the essence of proposition 3. It was proposed next that the initial model of the object is a necessary and sufficient requirement for complete design in proposition 4. Proposition 5 forms a basis for knowledge representation as a task tree. Evaluation requires knowledge of purpose(functions) and knowledge of structure, and proposition 6 highlights this point.

rDPM proposes that one, two or more atom components integrate to produce the intermediate form. While integrating, it inducts the world knowledge in order to achieve the refined parent subcomponent or in other words, the intermediate subcomponent. rDPM identifies that the description of a particular subcomponent, while analyzing the initial model of the object differs considerably from synthesis of the same due to the obvious presence of inductive knowledge. rDPM advocates that the synthesis of the elements involves two steps: (1) the

**establishment of logical relations between two, three or more subcomponents at knowledge plane as well as at parametric plane and (2) the actions or operations to establish this relation(s). These two steps are illustrated in propositions 7 and 8 respectively.** During scrutiny and alter phase of design, failure if any is thoroughly analyzed using domain knowledge and if needed modification process is recommended. This process runs recursively till the component meets the required specifications.

Hence, through rDPM, the study proposes a paradigm **select-decompose-integrate-scrutiny-alter** for design activity at representative level.

Chapter 3 devotes itself to model the cognitive processes involved in **the** human designer. The model is termed as cognitive Design Process Model(cDPM). The study examines the actions of the human expert' and models his behavior.

The advantage taken by distinguishing between logical and structural is that the word logical encompasses the word structural in addition to the incompleteness that prevails in the design activity in general. cDPM views design of an object as a logical entity and its decomposition leads to logical sub-entities and further into logical elements which are similar to subcomponents and atom components of rDPM. This mechanism is deductive. The project identifies **that** atleast a logical principle / truth is required to characterize these logical meta-entities found interspersing by logical bridges/connectors/mechanism. The cyclic abductive and inductive mechanisms bring the knowledge about meeting the essential qualities( $F^e$  elements) and desired qualities(number of  $F^d$  elements + operations regarding them to augment with available subcomponents) respectively and integrate with the current subcomponent resulting in its parent subcomponent. The quality of subcomponent is tested by constraint satisfaction as a basis, using en-

gineering analysis. This process is deductive. The alteration process is treated using abductive and inductive mechanisms.

Finally, the design process is tried to understand and model at system level in chapter 4. At this context, how design progresses at Systems Engineering level i.e. **design life cycle model** is suggested. Also, design progress is compared between system level description and representative level description. The model reveals that the design process involves three phases; **planning, scheduling and implementation** . During planning, the designer knows what **components** he has to design in addition to the availability of the relevant knowledge. This is akin to a combination of select and decomposition phases of rDPM. At system level, scheduling is very important because component hierarchy must be available before the initiation of actual design.

In order to prove the understanding of design at system level, an experimental expert working-system called **X-DAPS** is developed, based purely on the conviction that an expert system is an integrated program of knowledge base and inference mechanism with a sufficient separation. *X-DAPS* is an expert design assistant for DC Power Supplies, which is developed in 'C language. The kinds of knowledge, their manipulation and constraint satisfaction methods etc proposed in DPT and consequently in rDPM are observed while system development and validated from the experimentation on real world problem like DC Power Supplies design. It was observed that design of Power Supplies task offers sufficient generality to be as an example. Later, two more examples, one from mechanical engineering and another from chemical engineering are considered, chiefly to validate inductively the unification of models of design at all levels: the representative, the cognitive and the system, in addition to show that design itself is a generic activity.

**Screw Jack design**, an illustration from mechanical engineering is a case of complexity intensive design whereas **Heat Exchanger design**, a case from chemical engineering is inconsistency dominated design. In either case, it was proved that rDPM is sufficiently general to represent all these types of task solutions and the features of *X-DAPS* are adequate to represent any type of design from any type of domain. In other words, understanding design tasks suggest that their characteristics differ in degree than in content.

In a nutshell, it can be suggested that design possess certain generic nature and due to the degree of difference in their characteristics, different domains with respect to design appear different in their outlook. By observing what humans do, what different domains say, the present thesis proposed representative, cognitive and system level models and found out the unification in problems and problem solving techniques.

## 5.2 Future Directions

Obviously, there is no study that is said to be conclusive or **completed** in a subjective sense and the present study is in no way exception to this. The following directions are contemplated to the present study as futuristic.

1. **Design is a process where online acquisition of knowledge and on-line organization is a compulsion rather than exception.** Knowledge acquisition and organization is no easy job and generally is a bottle neck in development of expert systems or knowledge engineering tools for CAD.
2. **Design tasks need a constant user/designer interaction especially**

**for supplementing subjective judgments.** This leads to have a highly user-friendly front end with integration of graphic modules. This is also important when both system and user could not take any decision and user needs figurative images to decide.

3. **Prediction tools are required to be developed for efficient decision making process.**
4. **There exists in the present design theory proposed, a basic structure for intelligently supporting the innovative design.** However, they need to be exploited at the system level.
5. **At present, the system assumes that past experience i.e., knowledge that come from past experience is available.** However, this may not be the case always. Hence, built-in learning modules are required to capture the past experience.
6. **Though, the present work envisages the significance of the role of operations in design, however, this is not seriously looked into.** One of the reasons is the hypothesis that operations pose additional constraints to design synthesis and no way hinders understanding design process. But, in any case, an exhaustive study is needed at system level. Perhaps, this is one of the immediate tasks the thesis recommends.

## **Functions, Constraints and Components of DC Power Supplies**

In this appendix, the knowledge about functions, constraints and components involved in designing a specific type of DC power supply is outlined. The design methodology and the concerned decomposition tree are given in chapter 4 (section 4.3, 4.4).

Functions	Constraints	Components
<p>1. Primitive model:</p> <p>a) conversion of AC to DC</p>	<p>1. Application (receiver etc.,)</p>	<p>DC1( A specific type of DC power supply)</p>
<p>2. DC1 Power Supply:</p> <p>a) step down of voltage</p>	<p>1. power rating is small 2. leakage reactance is to be minimum</p>	<p>Small Power Shell Type <b>Transformer</b> (SSPT)</p>
<p>b) converting AC supply to Uni-directional</p>	<p>1. higher efficiency 2. less ripple 3. more conduction 4. less peak inverse voltage</p>	<p>Bridge rectifier</p>
<p>c) smoothening the unidirectional AC waveform and to make the output voltage free from ripple</p>	<p>1. more load current 2. regulation is to be more 3. ripple is independent of load resistance</p>	<p>LC filter</p>
<p>3. Small Power Shell Type Transformer:</p>		
<p>a) magnetic flux to be carried</p>	<p>1. good permeability 2. to reduce eddy current losses 3. reliability and low cost</p>	<p>Core (Laminated Steel Sheets)</p>
<p>b) provide mechanical support to windings</p>		
<p>c) to transfer electrical power from primary circuit to secondary circuit with given voltage to required voltage</p>	<p>1. good electrical conductivity 2. medium cost 3. good winding material</p>	<p>Windings (<b>Copper</b>)</p>
		<p><i>contd...</i></p>

Functions	Constraints	Components
<p>4. Bridge rectifier:</p> <p>a) To conduct on alternate half cycles</p> <p>5. windings:</p> <p>a) to feed input electrical power</p> <p>b) The power with transformed voltage to be taken</p> <p>6. LC filter:</p> <p>a) low impedance to be offered to AC components</p> <p>b) to remove the varying component in rectifier output</p> <p>c) proper functioning of the LC circuit</p>	<p>1. low forward resistance</p> <p>2. efficiency is to be more</p>	<p>Si-diode</p> <p>Primary Winding</p> <p>Secondary Winding</p> <p>Inductor (L)</p> <p>Capacitor(C)</p> <p>Bleed Resistor(<math>R_B</math>)</p>

## Design of Screw Jack System

### B.1. Introduction

The unique function of Screw Jack is to lift the loads. As was mentioned earlier, Screw Jack is a kind **amongst** earliest, Screw Jack as a kind amongst several kinds of lifting machine.

A typical Screw Jack principally consists of a lifting mechanism and a supporting mechanism. The lifting mechanism is called the Screw & Nut part of Screw Jack whereas the latter part is called the body of the Screw Jack. Avoiding **the** toppling of the load (the object subject to lift) is a primary concern in designing the body besides the low cost, portability and adaptation to various kinds of floors. In order to avoid toppling, for softer floors, a large base is used. On the other hand, a smaller base **has** advantages of portability and low cost. Hence a judgment is made depending on the environment in which the object under design (Screw Jack) is employed.

The lift mechanism or Spindle-nut system has to satisfy atleast four functions: lifting the load, preventing the rotation of load, transforming the **workdone**(effort) by the operator to the Screw and finally preventing Screw to completely release from the nut. Various components fulfill these functions respectively as Screw & Nut, loading platform and circular support endface of Spindle, handle & head of Spindle, end stopper washer & Screw.

The Table B.1 describes the different subcomponents and atom **components** and their

respective functions. However, these are not arranged in a sequence as that of proposed design. Fig. 4.10 shows this arrangement in a tree form after shuffling according to design process at system level. Fig. 4.9 indicates the overall structure of **the** object and its split drawing.

subcomponent	atom component	functions to be fulfilled
1. Screw & Nut	a. Screw	1. lifting the load through rotation
	b. Nut	2. supporting the screw and causing linear motion to the screw
2. loading plat form (cup)	a. Loading cup	1. supports the load 2. prevents rotation of load
	b. Supporter on spindle	1. transfer the gravity force of load to screw 2. offers smooth interface for loading cup
3. Handle & Head of the spindle	a. Handle	1. reduces the effort <b>applied</b> by operator
	b. Head of the spindle	1. transfer the effort of operator to the screw
4. End stopping	a. Washer	1. allows screw rod to travel up only till nut's bottom
	b. Bolt	1. holds washer in <b>position</b>
5. Support system	a. Body	1. provides support base for entire jack 2. accommodates nut & screw system 3. avoids toppling
	b. Grub screws	1. prevent rotation of nut 2. prevent upward movement of nut

Table B1

## B.2 Comprehensive Design of Screw Jack

**Problem inputs:** Load(W) and Lift(L)

### • 1. Lift Causing Mechanism

#### - 1.1. Design of Spindle

Thread type: Square

##### \* 1.1.1. Design of Screw

###### [Step 1.]

Calculate core diameter of Screw ( $d_{1,s}$ )

$$W = \frac{\pi}{4} d_{1,s}^2 \frac{f_{c,s}}{k_s}$$

$f_{c,s}$  - Ultimate compressive stress of material read from tables.

$k_s$  - Factor of safety of spindle

Check, whether  $d_{1,s}$  confirms standard size in tables. If not, select nearest higher size. Then select corresponding pitch ( $p_s$ ) of higher range.

###### [Step 2.]

Calculate outside diameter of screw ( $d_{2,s}$ )

$$d_{2,s} = d_{1,s} + p_s$$

###### [Step 3.]

Helix angle of screw

$$(\alpha) = \tan^{-1} \left( \frac{p_s}{\pi d_m} \right)$$

where Mean diameter of Screw

$$d_m = \frac{d_{1,s} + p_s}{2}$$

Check, for self locking :

$$\phi > \alpha$$

where  $\phi$  is the friction angle.

$$(\phi) = \tan^{-1} \mu_1$$

$\mu_1$  is coefficient of friction between Spindle and Nut materials that read from tables.

If not, select lower range of pitch( $p_s$ ) or next higher  $d_{1,s}$  or combination of both and repeat step 1.2. **until** this condition satisfies. Further this  $d_{1,s}$  and  $p_s$  are checked using stress criteria.

Where

$$\frac{\text{ultimate - shear - stress}}{\text{induced - max - shear - stress}} = \frac{f_{s,s}}{if_{smax}} \geq K_s$$

The ultimate-shear-stress and induced-max-shear-stress are evaluated as follows:

Calculate frictional torque required to overcome friction at the thread surface ( $T_1$ )

$$T_1 = W \frac{d_{1,s}}{2} \tan(\alpha + \phi)$$

Calculate induced torsional shear stress ( $if_{s,s}$ )

$$T_1 = \frac{\pi d_{1,s}^3}{16} if_{s,s}$$

Calculate induced direct compressive stress ( $if_{c,s}$ )

$$W = \frac{\pi d_{1,s}^2}{4} if_{c,s}$$

Calculate maximum principle shear stress induced **due** to combined direct compressive and torsioned shear stress ( $if_{smax}$ )

$$if_{smax} = \frac{if_{c,s}}{2} + \sqrt{\left(\frac{if_{c,s}}{2}\right)^2 + (if_{s,s})^2}$$

In case the above criteria is not satisfied, next higher  $d_{1,s}$  and corresponding  $p_s$  from tables are selected and the procedure is repeated from step 1.2.

#### [Step 4.]

Calculate the length of the threaded portion. Screw ( $l_{1,s}$ )

$$L_{1,s} = L + l_n + a_1$$

L - Lift

$l_n$  - length of nut

$a_1$  - extra length = 5 mm

Since  $L_{1,s}$  is dependent on length of nut, here length of the nut has to be designed. However, the complete design of nut is found elsewhere later as an independent component. i.e., here length of the nut can be treated as a constraint for finding length of the screw. The following is the procedure to find the length of the nut.

core diameter of nut thread

$$(d_{1,n}) = d_{1,s}$$

major diameter of nut thread

$$(d_{2,n}) = d_{2,s}$$

pitch of nut thread

$$P_n = p_s$$

Calculate number of threads (N) in nut

$$W = N \frac{\pi}{4} [d_{2,n}^2 - d_{1,n}^2] \times \frac{f_{b,n}}{k_n}$$

$f_{b,n}$  - ultimate bearing stress of nut material( read from tables)  $k_n$  - factor of safety for nut

N is to be corrected to nearest higher integer.

Length of nut

$$l_n = N \times p_n$$

$$\text{thus, } l_{1,s} = L + l_n + a_1$$

## 1.1.2. Design of End Stopper

### 1.1.2.1. Design of End Bolt

[Step 1.] Spindle end bolt diameter

$$(d_{1,b}) = \frac{d_{1,s}}{5}$$

[Step 2.] Length of bolt ( $l_{1,b}$ ) — 15 mm

[Step 3.] specification of bolt : hexagon bolt

$$M = d_{1,b} \times 15$$

as per Table IS : 1364

### 1.1.2.2. Design of End Washer

[Step 1.] Spindle end washer inner diameter

$$(d_{1,w}) = d_{2,s} + 0.5mm$$

[Step 2.] Spindle end washer outer diameter

$$(d_{2,w}) = d_{2,s} + 15mm$$

[Step 3.] Thickness of Washer ( $l_{1,w}$ ) — 3 mm

[Step 4.] Size of threaded hole at the end of spindle

$$(d_{6,s}) = d_{1,b}$$

$$l_{5,s} = l_{1,b}$$

[Step 5.] Specification : M ( $d_{6,s}$ ) × 15 deep

## \* 1.1.3. Design of Head Stock

### 1.1.3.1. Design of Pin

[Step 1.] Pin diameter

$$(d_{3,s}) = \frac{d_{4,s}}{4}$$

where  $d_{4,s}$ , the Spindle head diameter

$$(d_{4,s}) = 1.7d_{2,s}$$

[Step 2.] Length of Pin

$$(l_{2,s}) = \frac{d_{4,s}}{8}$$

### 1.1.3.2. Design of Handle

[Step 1.] Total length of handle

$$l_{1,H} = l_H + \frac{d_{4,s}}{2} + H$$

where  $l_h$ , the distance from the center of the spindle to effort application point on handle ( $l_H$ ), is calculated from

$T = n p l_H$  and

T - total Torque to be applied is calculated as follows:

Calculate frictional torque between loading cup and spindle head ( $T_2$ )

$$T_2 = \frac{2}{3} \mu_2 W \left[ \frac{\left(\frac{d_{4,s}}{2}\right)^3 - \left(\frac{d_{3,s}}{2}\right)^3}{\left(\frac{d_{4,s}}{2}\right)^2 - \left(\frac{d_{3,s}}{2}\right)^2} \right]$$

$\mu_2$  = coefficient of friction between spindle and loading

$$T = T_1 + T_2$$

n - number of operators of Screw Jack and effort applied by each person on handle(p)

H is Extra length for hand grip (constant)

Check, whether  $l_{1,H}$  is 600 to 1000 mm

if  $l_{1,H}$  is less than 600 reduce the effort p and repeat step 1.

if  $l_{1,H}$  is greater than 1000 increase the effort (user option) by small amount and repeat step 1.

Still, if  $l_{1,H}$  is greater than 1000 consider number of operators n higher value and repeat step 1.

[Step 2.] Calculate diameter of handle ( $d_{1,H}$ )

$$M = \frac{\pi}{32} d_{1,H}^3 \frac{f_{b,H}}{k_H}$$

$f_{b,H}$  - ultimate bending (tensile) stress of handle material (read from tables)

$k_H$  - factor of safety for handle where maximum bending moment on handle

$$M = Pl_e$$

and the effective length of handle ( $l_e$ ) used in handle bending effect

$$l_e = l_H - \frac{d_{4,s}}{2}$$

### 1.1.3.3. Design of Head

**[Step 1.]** Two holes of diameter  $d_{s,s}$  at right angles to each other are bored in the spindle head for the convenience of changing the position of handle after every quarter revolution. Calculate diameter of holes ( $d_{s,s}$ ) in the spindle head from

$$D_{5,s} = d_{1,H} + a_2$$

$a_2$  - allowance for free handle entry

Check, if

$$\frac{\text{ultimate - compressive - stress}}{\text{induced - compressive - stress}} = \frac{f_{c,s}}{i f_{c,s}} \geq k_s$$

where the area of cross section of head ( $A_s$ ) through two holes

$$A_s = \left( \frac{\pi}{4} d_{4,s}^2 \right) - [(d_{1,H} d_{4,s}) + (d_{4,s} - d_{1,H}) d_{1,H}]$$

and the induced compressive stress ( $i f_{c,s}$ )

$$W = A_s i f_{c,s}$$

If the condition is not satisfied reduce  $d_{5,s}$ , select lower  $l_{1,H}$  in the permitted range and repeat from step 1.1.3.2. Still, if it is not satisfied select

$$d_{4,s} = 2d_{2,s}$$

and repeat from step 1.1.3.1.

**[Step 2.]** Length of head

$$l_{3,s} = 2d_{s,s}$$

When Screw Jack is in maximum lift condition, screw can be looked upon as a column fixed at the lower end and free at the

top and checked for buckling. The buckling or critical load is calculated as

$$W_{c,r} = \frac{\pi}{4} d_{1,s}^2 f_{y,s} \left[ 1 - \left( \frac{f_{y,s}}{4c\pi^2 E} \times \left( \frac{l_u}{k} \right)^2 \right) \right]$$

where  $l_u$  is the unsupported spindle rod

$$(l_u) = L + \frac{l_n}{2} + l_{3,s}$$

$f_{y,s}$  - yield stress of material (read from tables)

C - End fixity Coefficient = 0.25

k - radius of gyration = 0.25  $d_{1,s}$

and

$$\frac{\text{Buckling - load}}{\text{Design - load}} = \frac{W_{c,r}}{W} \geq k_{s,j}$$

$K_{s,j}$  = Factor of safety for Screw Jack = 4 (or from user)

#### \* 1.1.4. Design of Loading Cup

[Step 1.] Inner diameter

$$(d_{1,c}) = d_{3,s} + 1mm$$

[Step 2.] Cup seat diameter

$$(d_{2,c}) = d_{4,s}$$

[Step 3.] Thickness

$$l_{2,c} = l_{3,c} = \frac{d_{2,c}}{5}$$

[Step 4.] Diameter

$$d_{3,c} = 1.5d_{2,c}$$

#### - 1.2. Design of Nut

[Step 1.]

$$d_{1,n} = d_{1,s}$$

[Step 2.]

$$d_{2,n} = d_{2,s}$$

[Step 3.] Length of nut from Step 1.1.1.

$$(l_n) = l_{1,n} + l_{2,n}$$

[Step 4.] Calculate outer diameter of the nut body ( $d_{3,n}$ )

$$W = \frac{\pi}{4} [d_{3,n}^2 - d_{2,n}^2] \times \frac{f_{t,n}}{k_n}$$

$f_{t,n}$  - ultimate tensile stress for nut material (read from table)

$k_n$  - Factor of safety for nut

[Step 5.] Calculate diameter of collar ( $d_{4,n}$ )

$$W = \frac{\pi}{4} [d_{4,n}^2 - d_{3,n}^2] \times \frac{f_{c,n}}{k_n}$$

$f_{c,n}$  - ultimate crushing stress for nut material(read from table)

[Step 6.] Calculate the thickness of collar ( $l_{1,n}$ )

$$W = (\pi d_{3,n}) \times (l_{1,n}) \frac{f_{s,n}}{k_n}$$

$f_{s,n}$  - ultimate shear stress for nut material(read from table)

## • 2. Design of Supporting Mechanism

Supporting mechanism of screw jack consists of a body and a grub screw.

### - 2.1. Design of Body

[Step 1.] Inner diameter

[Step 2.] Outer diameter

$$(d_{2,B}) = 1.5d_{4,n}$$

[Step 3.] Base Inner diameter

$$(d_{3,B}) = 2.25d_{4,n}$$

[Step 4.] Base outer diameter

$$(d_{4,B}) = 1.75d_{3,B}$$

[Step 5.] Length of body

$$(l_{1,B}) = l_{1,n} + L + 35\text{mm}$$

[Step 6.] Length of collar

$$(l_{2,B}) = l_{2,n}$$

[Step 7.] Thickness of body

$$(l_{3,B}) = 0.25d_{2,s}$$

[Step 8.] Thickness of base flange

$$(l_{4,B}) = 2l_{3,B}$$

[Step 9.] Center distance of Grub screw here from top face

$$(l_{5,B}) = \frac{l_{2,B}}{2}$$

[Step 10.] Outer diameter of threaded hole for grub screw

$$(d_{5,B}) = \frac{l_{2,B}}{5}$$

with a minimum of 5 mm

## — 2.2. Design of Grub Screw

[Step 1.] Length of grub screw

$$(l_{1,G}) = \frac{d_{2,B} + d_{1,B}}{2}$$

[Step 2.] Outer diameter of grub screw

$$(d_{1,G}) = d_{5,B}$$

[Step 3.] Specification: Slotted head grub screw

as per IS: 2388 Code Table

NOTE : All the dimensions, including diameters and lengths can be read from the part drawing (Fig. 4.9). Here, the second subscripts c, s, n, H, B, w, b and g denote respectively the loading cup, spindle, nut, handle, base, washer, bolt and grub screw.

## Design of 1,2 shell & tube Heat Exchangers

### C.1 Introduction

Shell & tube Heat Exchangers are primarily used to transfer heat thereby increase or decrease either temperature or heat content or both between any two fluids. The fluid having more significance is called process **stream** or process fluid whereas the other one is called cooling or heating stream or fluid.

In order to fulfill this unique of Heat Exchanger, two principal structural components are envisaged: tubes and shell. There exists a criteria to allocate which fluid has to flow in which side (tube side or shell side). In case of thermal design, more attention is devoted to heat transfer aspects. That is the parameters that directly related to efficient heat transfer. However, complete structural description, which includes mechanical aspects, such as supports is not covered. In order to design tube bundle, about five parameters have to be evaluated, for which procedures are available. Similarly for shell design, four parameters need to be found out, among which one parameter is infact a cluster of few other parameters (base parameters).

Fig. 4.11 shows the diagram of 1,2 shell & tube Heat Exchanger and Fig. 4.12 represents component tree/task tree. *During* design the user is expected to provide the following inputs.

1. Fluid quantities (W, w)

2. Physical properties of the fluids (  $c, k, \mu, \rho$  )
3. Ingoing and outgoing temperatures
4. Operating pressures (absolute)
5. Allowable pressure drops
6. Fouling factors
7. corrosion allowance
8. size / space limitations

#### C.2 Comprehensive Design of 1,2 Shell & Heat Exchanger

As a prelude, allocate fluids that flow in tubes and in shell. Considering, the constraints like corrosion, **fouling**, temperatures and pressures, pressure drops and viscosity, the fluids will be allocated to flow into either tube side or shell side.

• 1. Design of Tube bundle

- 1.1. Finding the tube OD and BWG

Step 1.

$$LMTD = \frac{(T_1 - t_2) - (T_2 - t_1)}{\ln \left( \frac{T_1 - t_2}{T_2 - t_1} \right)}$$

Step 2. Temperature correction factor,  $F_T$  is calculated/ read from graph given LMTD and kind of heat exchanger

Step 3.

$$\Delta T_{mean} = LMTD \times F_T$$

Step 4.

$$r = \frac{T_1 - T_2}{t_2 - t_1}$$

$$k_c = \frac{U_S - U_T}{U_S} = \frac{U_{shell-side} - U_{tube-side}}{U_{shell-side}}$$

approximate values of  $U_S$  &  $U_T$  can be read from tables for given heat exchanger type and type of fluid.

Find  $F_c$  from figure given  $r$  and  $K_c$  values.

$$T_c = T_2 + F_c(T_1 - T_2)$$

$$t_c = t_1 + F_c(t_2 - t_1)$$

Evaluate all properties  $c, \mu, k, p$  at this corrected **temperatures**

Step 5. Heat load

$$Q = WC(T_1 - T_2) = wc(t_2 - t_1)$$

from this, sometimes the one parameter which is not given can be found out. Quite often, the problem gives free hand to the designer to fix the flow rate or outlet temperatures of the cooling / heating stream.

**Step 6. Let  $U_D = U_T$**

net area of contact (total HT area),

$$A = \frac{Q}{U_D \Delta T_{mean}}$$

**Step 7.** Assume  $D_0$  and BWG (Gauge), L of each tube from standards. (knowledge is available for guidance to the user)

1. Generally  $D_0$  is in between (5/8)" to (1.5)"

In case of lower  $D_0$ , HT coefficient reduces, surface area per tube increases and pressure drop increases. Converse is the effect of higher  $D_0$ . So, a judgment has to be made for assuming  $D_0$ .

2. For higher operating pressures and lengthy pipes BWG is more.
3. If space restriction is given, take L as length given by problem description, otherwise, L is standard length. If designer feels the standard length is too lengthy, he can assume length, keeping flow rates and pressure drops in view. Lengthy tubes increase the pressure drop, occupy more space, increase number of supports for countering bending and sometimes influence the choice of flow pattern (counter current flow and cross flow:  $\zeta_n$  the former HT coefficient is more than that of later).

- **1.2. Finding the tube** geometric arrangement and pitch

\* **1.2.1. Finding tube geometric arrangement** based on judgment. There exists two conventional geometric arrangements, triangular and square types. However, the user depending on his requirement can have other arrangements also.

In case of triangular arrangement, the constraints will be the

1. better HT coefficient in shell side
2. providing higher HT area for a given shell diameter
3. cleaning is difficult

On the other hand, in case of square pitch, the constraints will be the

1. cleaning is easier
2. leads to low pressure drop
3. results in low HT coefficient

(for exhaustive information see **Kern 1950; Perry and Chilton 1983; Fanaritis and Bevevino 1979; Lord et al., 1979; Yokell, 1990; Guyer et al., 1989** ).

\* 1.2.2. Finding the pitch

$$P_T = D_0 \left( 1 + \frac{c'}{D_0} \right)$$

or

$$P_T = (1 + x)D_0$$

x is between 25 % - 30 %

$c'$  = clearance between two adjacent tubes in a unit geometric arrangement

x = margin of safety that subsumes clearance values

- **1.3. Finding number of tubes**

surface-area/tube =  $\pi D_0 L$

$$n' = \frac{\text{total} - HT - \text{area}}{\text{surface} - \text{area}/\text{tube}} = \frac{A}{\pi D_0 L}$$

Number of tubes calculated,  $N_{cal}$  = nearest higher integer of  $n'$

Exact number of tubes estimated =  $N_t$  is read from table giving  $D_0$ ,  $P_T$ , number of passes in tube side. Number of passes, n is 2 in present case but this is always not essentially constant.

- **1.4. Finding heat transfer coefficients,  $h_i$  &  $h_o$**

**Step 1.** Find Flow area of each tube =  $a_t'$  from given Tube OD & DWG

**Step 2.** actual flow area

$$a_t = \frac{N_t a_t'}{144n}$$

Step 3. mass velocity

$$G_t = \frac{W}{a_t}$$

Step 4.

$$Re_t = \frac{D_i G_t}{\mu}$$

Step 5. find  $J_H$  at  $Re_t$  from figure (graph)

**Step 6. find  $\frac{C\mu}{k}$**

**Step 7.**

$$h_i = J_H \phi_t \frac{k}{D_i} \sqrt[1/3]{\left(\frac{C\mu}{k}\right)}$$

**Step 8.**

$$h_{i_o} = h_i \frac{D_i}{D_o}$$

initially, at  $\phi_t = 1$ ,  $h_i, h_{i_o}$  are estimated

Step 9. From this the Wall temperature is calculated

$$t_w = T_c - \left(\frac{h_o}{h_{i_o} + h_o}\right) (T_c - t_c)$$

$h_o$  comes from shell design. Assume  $h_o = h$ , for further calculations and when shell design is completed read  $h_o$  and redo **the** entire design.  
thus

$$\mu_w = \mu \quad \text{at } t_w$$

$$\phi_t = \left(\frac{\mu}{\mu_w}\right)^{0.14}$$

from this  $\phi_t$  again find  $h_i, h_{i_o}$

- **1.5. Finding Pressure Drop**

**Step 1.** Find friction factor  $f$  given  $Re_t$  from graph

**Step 2.** Pressure drop due to Friction loss,

$$\Delta P_t = \frac{f G_t^2 L n}{5.22 \times 10^{10} D_i \rho_t \pi_t}$$

**Step 3.** Pressure drop due to return loss,

$$\Delta P_r = \frac{4nV^2}{2g\rho_t}$$

**Step 4.** Total pressure drop,

$$\Delta P_{est,t} = \Delta P_t + \Delta P_r$$

check if

$$\Delta P_{est,t} > P_{allowed}$$

in case the condition is satisfied, design results be declared.

In case of alteration, whether a modification in  $D_0$  has any improvement has to be checked. As a next step the number of passes are altered and the calculations are repeated till the condition satisfy. Notice that number of passes are always positive integers.

• **2. Design of shell**

- **2.1. Finding the inner diameter,  $D_{i,s}$ , of shell**

Find shell inside diameter of shell,  $D_{i,s}$  from table given  $D_0$ ,  $P_T$ ,  $n$ ,  $N_t$  of tube side.

- **2.2. Design of Baffles**

\* **2.2.1. Finding Baffle type and Baffle cut**

**Step 1.** Baffle selection

Types 1. Horizontal cuts

2. Vertical cuts

**Window openings** 1. large

2. low

**Segmented baffles** 1. good turbulence

2. good HT

**Criteria:** (generally horizontal baffles for sensible-heat exchangers.)

**Step 2.** Finding baffle cut between 20 % - 25 % (user **selection**)

- **2.2.2. Finding number of baffles,** Number of baffles,  $N_B$  - 5 in general or user given

\* **2.2.3. Finding baffle-baffle distance (baffle space)** Baffle space,

$$B = \frac{D_{is}}{\text{number of baffles}} = \frac{D_{is}}{N_B}$$

### 2.3. Finding Shell side HT coefficient

**Step 1.** Flow area,

$$a_S = \frac{D_{is} c B}{144 P_T}$$

**Step 2.** Mass velocity,

$$G_S = \frac{W}{a_S}$$

**Step 3.** Equivalent diameter (for square Pitch),

$$D_e = \frac{\left( P_T^2 - \frac{\pi d_o^2}{4} \right)}{\left( \frac{\pi d_o}{4} \right)}$$

Similarly for a given Pitch type  $D_e$  is found.

For a triangular Pitch,

$$D_e = \frac{\left( 0.43 P_T^2 - \frac{\pi d_o^2}{8} \right)}{\left( \frac{\pi d_o}{8} \right)}$$

Step 4. Reynold's number,

$$Re_S = \frac{D_e G_S}{\mu_S}$$

Step 5. Read  $J_H$  given  $Re_S$  from graph

**Step 6.** HT coefficient,

$$h_o = J_H \phi_S \left( \frac{k_S}{D_e} \right) \sqrt[3]{\left( \frac{c_S \mu_S}{k_S} \right)}$$

#### 2.4. Finding pressure drop at shell side

Step 1. find  $f$  from  $Re_S$  (graph)

**Step 2.** Number of crosses,

$$N + 1 = 12 \left( \frac{L}{B} \right)$$

Step 3.

$$\Delta P_{S,est} = \frac{f G_S^2 D_S (N + 1)}{5.22 \times 10^{10} D_e \rho_S \phi_S}$$

check if

$$\Delta P_{est,S} > \Delta P_{allow}$$

In case the condition is not satisfied, reduce baffle spacing by increasing number of baffles and find new  $\Delta P_{est,S}$ . This procedure is repeated until the condition is satisfied.

Now thermal design of heat exchanger is said to be completed with respect to pressure drop criteria. A parameter, often, the designer have to provide is the overall dirt factor which indicates the requirement of cleaning the tubes.

- **3. Finding the Dirt factor**

**Step 1.** Clean HT coefficient,

$$U_c = \frac{h_{i_o} h_o}{h_{i_o} + h_o}$$

**Step 2.** dirt factor,

$$R_D = \frac{U_C - U_D}{U_C U_D}$$

If  $R_D = 0 \rightarrow$  **No heat transfer** is possible, i.e. the tubes require cleaning.  
when  $R_{D,deposited}$  is greater than  $R_{D,allowed}$  it requires cleaning.

LMTD		Log mean temperature difference
T		Temperature of the cold fluid
$t_1$		Temperature of the hot fluid inlet condition
$t_2$		temperature of the hot fluid outlet condition
$t_c$		temperature of the hot fluid corrected value
r, $K_c$		Constants
U		Overall heat transfer coefficient
$\omega$	flow	rate of tube side liquid
$C_c$		Specific heat of shell side & tube side
$D_0$		outer diameter of the tube
BWG		British Wire Gauge
L		length of each tube
HT		Heat Transfer
HT Coe		Heat Transfer coefficient
$N_t$		Number of tubes
$\delta P_{est,t}$		Pressure drop estimated at tube side
n		number of tube passes(In this design n = 2,) however, this depends on pressure drop constraint)
$Re_t$		tube side Reynold's number
$D_i$		read from table given $D_0$ & BWG(thickness)
$h_{io}$		HT coefficient in the tube at wall
$h_i$		inside HT coefficient in tube
$J_H$		colburn's value
k		thermal conductivity
$\mu$		viscosity
$\phi_t$		viscosity correction due to temperature factor, $\left(\frac{\mu}{\mu_w}\right)^{0.14}$
$\rho_t$		density at $t_c$
v		linear velocity
g		acceleration due to gravity
ID		inside diameter
$\Delta P_{est,S}$		estimated shell side pressure drop
$\Delta P_{S,allowed}$		allowed shell side pressure drop
$C_t$		tube-tube clearance given or find from $P_t - D_0$
M		mass now rate
$\mu_S$		at $t_c$
$c_S$		at $t_c$
$k_S$		at $t_c$
$\rho_S$		at $t_c$
$\phi_S$		Viscosity-correction = $0.14 \sqrt{\left(\frac{\mu}{\mu_w}\right)} \mu_w$ at $t_w$

subscript S	shell
subscript T	tube
subscript cal	calculated value

## REFERENCES

- Akman, V. et al. (1990) Knowledge engineering in design, Knowledge-Based Systems, Vol 1 No 2 March 1988 pp. 67-77,
- Akin, O., (1986)** 'Psychology of Architectural Design', Pion, London, U.K., (1986)
- Alexander, **Tom., (1982)** 'Computers on the road to self-improvement', Business Week, pp 148-160, June 14, (1982)
- Bahrami, A., (1992)** 'Designing Artificial Intelligence Based Software', Sigma Press-Wilmslow, (1992)
- Barret, **W., (1979)** 'The illusion of technique', Anchor Books, NY, (1979)
- Bell, K J., (1983) 'Thermal Design Of Heat Exchangers', in Heat Transmission, Chapter 10, Ed Knudsew, JG an Hand Book of Chemical Engineering, Fifth Edition, eds. Perry, R.H and Chilton, C.H., International Students Editions, Mc Graw-Hill, (1983)
- Bennett, J., (1978) 'SACON : A Knowledge-Based Consultant for Structural Analysis', Stanford University, Comp.Sc.Dept., Report No STAN-CS-78-699, (1978)
- Black, **Paul H and Adams, Eugene O., (1988)** 'Machine Design', McGraw-Hill International Editions, (1988)
- Boyle, **J M., (1989)** 'Interactive engineering systems design: a study for artificial intelligence applications', Artificial Intelligence in Engineering. Vol 4 No 2 pp 58-69, (1989)
- Bremdal, Brent,A., and Kristiansen Svein., (1986) 'LIFT: Concept Definition in Marine System Design', Application of AI in Engineering, 1st international conference, Southampton University, U.K., pp 403 - 421, eds. Sriram,D and Adey,R, (1986)
- Brown, **D.C and Chandrasekharan, B., (1983)** 'An approach to expert system for mechanical design', proc. trends and applications, Gainthersburg, MD pp 173-180, (1983)

- Brown, D.C. and Chandrasekharan, B., (1985)** 'Expert System for a Class of Mechanical Design Activity: Knowledge Engineering. In Computer-Aided Design, ed. Gero, J. S. pp 259-283, Amsterdam: North Holland, (1985)
- Brown, D.C. and Chandrasekharan, B., (1986)** 'Knowledge and control for a mechanical design expert system', IEEE Computer vol 19 pp 92-100, July (1986)
- Brown, D.C. and Robert Breau., (1986)** 'Types of Constraints in Routine Design Problem Solving', Applications of AI in Engineering problems, proceedings of the I International conference, Southampton University, U.K, eds. Sriram D and Aday R, A computational Mechanics publication, Vol 1, pp 383-390, (1986)
- Brown, D.C. and Chandrasekharan, B., (1989)** 'Design Problem Solving: Knowledge Structures and Control Strategies' San Mateo, Calif: Morgan Kaufmann, (1989)
- Chandrasekharan, B., (1990)** 'Design problem solving: A task *Analysis*' AI Magazine, pp 59-71, Winter (1990)
- Charniak, K., and Mc Dormott, D., (1985)** 'Introduction to Artificial Intelligence', Addison Wesley, (1985)
- Coyne, R., (1990) 'Design reasoning with out explanations', AI Magazine, pp 72-80, (winter 1990)
- Coyne, R.D. and Postmus, A G., (1990) 'Special Application of neural networks in Computer Aided Design', Artificial Intelligence in Engineering, Vol 5, No 1, pp 9-22, (1990)
- Dixon, J.R, and Simmons, M.K., (1984)** 'Expert Systems for Engineering Design: Standard V-Belt Drive Design as an Example of the Design-Evaluate-Redesign Architecture', ASME Computers in Engineering Conference, (1984)
- Duda, R.O, Hart, P.E, Konolige, K and Reboh, R., (1979)** 'A Computer-Based Consultant for Mineral Exploration', Tech. rep., SRI international, (1979).
- Duda, R.O, Hart, P.E, Konolige, K et al (1978)** 'Development of the PROSPECTOR consultation system for mineral exploration final report', SRI Projects 5821 and 6415, SRI International Inc., Mento park, calif, (1978)

- Dyer, M.G, Flowers, M and Hodges, J., (1986)** 'EDISON : An engineering design invention system operating naively', Applications of AI in engineering problems, vol 1, eds. Sriram D., Adey R., A computational **mechanics** pub., UK, pp 327-341 (Oct 1986)
- Eastman, **C.M., (1978)** 'The representation of design problems and maintenance of their structure, AI and pattern recognition in Computer Aided Design', ed. Latombe, North-Holland pub. co, (1978)
- Fanaritis, J.P and Benevino, J.W., (1979)** 'How to select the optimum shell-and-tube heat exchanger', in process heat exchange, ed. Cavaseno, V, Chemical Engineering, Mc Graw-Hill pub. co, NY, (1979)
- Feigenbaum., (1977)** 'PUFF: Pulmonary Function Diseases', (1977)
- Fenvis, S J, and Norabhoompipat, T., (1978)** 'AI and pattern recognition in Computer Aided Design', eds. Latombe, North-Holland pub co, (1978)
- Finger, S and Dixon, J R.,(1989)** 'A review of research in Mechanical Engineering Design. Part I: Descriptive, prescriptive and computer based models of design processes' Research in Engineering Design. Vol 1 no 1, pp 51-67, (1989)
- Fischler, M.A and Firschein, O., (1987)** Intelligence - The eye, The brain and the computer', Addison-Wesley pub. co, (1987)
- Forbus, K., (1983)** 'Intelligent Computer Aided Engineering', AI Magazine, (Fall 1988)
- Frost, Richard.,(1986)** 'Introduction to Knowledge Base Systems' MacMillan Publishing company, New York, (1986)
- Gazzania, M.S.,(1985)** 'The Social Brain', Basic Books, NY pp 134 (1985)
- Getmanova, A., (1989)** 'LOGIC, Progress Publishers, Moscow, (1989)
- Gero, J.S, (1987)** 'Towards a generalized model of diagnostic behavior ', Expert System Architecture, ed. Gero, J.S. pp 183-210, (1987)
- Gero, J.S., Maher, M.L. and Zhang, W., (1988)** 'Chunking structural knowledge as prototypes', J.S. Gero(ed.) Artificial Intelligence in Engineering: Design,3-21, Elsevier and Computational Mechanics pub., (1988)

- Gero, J S.,(1990)** 'Design prototypes: A knowledge representation schema for design **AI magazine**. pp 26-36, (Winter 1990)
- Gero, J.S and Rosenman, M A., (1990)** 'A conceptual framework for knowledge-based design research at Sydney University's design computing unit' **Artificial Intelligence in Engineering**. Vol 5 No 2, pp 65-77, (1990)
- Godwin, G.L., (1958)** 'Digital Computer Tap Out Designs for Large Motors ... ' **Fast. Power**, (April 1958)
- Guyur, E.C, and Bronwell, D.L ., (1989)** 'Hand book of Applied Thermal Design', **Mc Graw-Hill NY**, (1989)
- Harubs, Ed., Edwards, N., Nienou, M.F and Butterworths A.W., (1985)** 'Mixing in the process industries', (1985)
- Hall, A.S, Holwenka, A.R, Laughlin, H.G.,(1980)** 'Theory Problems of Machine Design', **Mc Graw-Hill, NY**, (1980)
- Haragu, S.S and Kusiak, A., (1987)** 'Analysis of Expert systems in manufacturing design', **IEEE Transactions on Systems, Man and Cybernetics**, Vol SMC -17, No 6, pp 898-912, (1987)
- Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., (1983)** **Building Expert Systems'**, Addison - Wesley Publishing Company Inc, Massachusetts, (1983)
- Howe, A. Cohen, P. and Dixon, J. (1986)** 'DOMINIC: A Domain-Independent Program for Mechanical Engineering Design', **Applications of AI in Engineering problems**, proceedings of the I International conference, Southampton University, U.K, eds. **Sriram D and Aday R**, A computational Mechanics publication, Vol 1, pp 289-299, (1986)
- Kolodner, J., (1989)** 'Judging which is the best case for a case- based reasoner', **Proceedings of the Case-Based reasoning workshop**, (1989)
- Kern, D Q., (1950)** 'Process Heat Transfer', **Mc Graw-Hill, Tokyo**, (1950)
- Levenspiel and Octave., (1972)** 'Chemical reaction engineering', **IInd Edtion**, Wiley Eastern Ltd (1972)
- Lindsay, R.K. etal., (1980)** ' Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project ', **New York , Mc Graw-Hill**, (1980)

- Lindsay, R.K, Buchanan, B.G, Feigenbaum, E.A, and Lederberg, J., (1980)** 'DENDRAL', New York: Mc Graw-Hill (1980)
- Lord, R.C, Minton, P.E and Slusser, R.P., (1990)** 'Design of Heat Exchangers' (1979)
- Lu, S.C-Y., Blattner, C.R., Lindem, T.J., (1986)** 'IDRILL: A Knowledge-Based Expert System for Drilling Station Design' Applications of AI in Engineering problems, proceedings of the I International conference, Southampton University, U.K, eds. Sriram D and Aday R, A computational Mechanics publication, Vol 1, pp 423-443, (1986)
- MacCallum, K J., (1990)** 'Does intelligent CAD exists?', AI in Engg Vol 5, no 2, pp 55-64, (1990)
- Maher, M.L, Fenves, J J., (1984)** 'HI-RISE: An Expert System for the preliminary Structural Design of High Rise Buildings', In Knowledge Engineering in Computer Aided Design, ed. Gero, J, IFIP, (Sep 1984)
- Maher, M L., (1989)** 'EDESYN: An Engineering Design Synthesis Environment', CAPE'89- Third International Conference on Computer Applications in Production and Engineering, Tokyo, Japan, (oct 1989)
- Maher, M L., (1990)** 'Process Models for Design Synthesis', AI Magazine pp 49-58, (Winter 1990)
- Maitra, G.M, Prasad, L.V., (1985)** 'Hand book of Mechanical Design' Tata Mc Graw-Hill, (1985)
- Mostow, T., (1985)** 'Towards Better Models of the Design Process', AI Magazine, Spring, pp 44-56 (1985)
- Mc Dermott, D., (1978)** 'Circuit Design as Problem-Solving', AI and Pattern Recognition in CAD, J.-C. Latombe, ed., North-Holland, Amsterdam, pp 227-245, (1978)
- Mc Dermott, D.,(1980)** 'R1: A Rule Based Configurer of computer systems', Carnegie-Mellon University, (1980)
- Mc Dermott, D., (1981)** 'R1: The formative years', AI magazine 2:21-29, (1981)
- Metin and Rezai, (1986)** 'An efficient method to predict the effect of design modification on the dynamics of structures', University microfilms International, Michigan (1986)

- Mittal, S and Araya, A., (1986)** 'A knowledge-based framework for Design', proc. AAAI-86 (1986)
- Naga Prasad, P and Reddy, A.S., (1992)** 'An overview on understanding design : an AI perspective', All India Conference on Expert Systems in Engineering pp 1.7.1-1.7.4, India,(January, 1992)
- Naga Prasad, P and Reddy, A.S., (1992)** 'The Role of Knowledge in Design Tasks', Innovative Applications in Computing, eds. Balagurusamy, E, and Sushila, B, pp 254-252 Tata McGraw-Hill, (1992)
- Naga Prasad,P., Reddy, A.S., and Kanaka Raju,P (1993)** 'Modeling Design Processes', revised version communicated to Knowledge-Based Systems, U.K., (1993)
- Nauman, E.B and Buffam, B.A., (1983)** 'Mixing in continuous flow systems', John Wiley & Sons, (1983)
- Nessler, H, and Stadelmeier, F, (1991)** 'Constructional Features and Characteristics of Transformers', Willy Eastern Limited, (1991)
- Neumann, Von, (1956)** 'Probabilistic logics and the synthesis of reliable organs from unreliable Components ', Automata Studies, No 34, (1956)
- Neveu, B and Haran, P (1986)** 'SMECI : An expert system for civil engineering design Applications of AI in engineering problems', vol 1, eds. Sriram D., Adey R., A computational mechanics pub., UK, pp 317-326, (Oct 1986)
- Newell, A., (1980)** 'Reasoning, Problem Solving, and Decision Process: The Problem space as a fundamental category. In Attention and Performance', Vol 8, pp 693-718, Hillsdale, N.J.: Lawrence Erlbaum (1980)
- Pandya and Shah, (1987)** 'Machine Design', 1987
- Penfield, W., (1978)** 'The Mystery of the Mind', Princeton University press NJ, (1978)
- Pericera, L., (1978)** 'AI Techniques in Layout Design', AI and pattern recognition in Computer Aided Design, eds. Latombe, North-Holland pub co, (1978)
- Perry and Chilton (1983)** 'Chemical Engineering Hand Book', Mc Graw-Hill pub., (1983)

- Pople, H.E, (1982)** 'Heuristic methods for imposing structure on ill structured problems: The structuring of medical diagnosis', AI in medicine, ed. Szolovits, P., pp 119 - 185, Colorado : Westview Press, (1982)
- Preiss, K, (1978)** 'In general discussion 2 in the IFIP TC-5 Working Conference on AI and Pattern Recognition in Computer Aided Design', Grenoble, France, (March 17-19, 1978)
- Purohit, G.P (1983)** 'Thermal and hydraulic design of hairpin and **finned-bundle** exchangers', Chemical **Engineering**. pp 62-70 (May,1983)
- Raina, K.B and Bhattacharya, S.K., (1991)** 'Electrical Design Estimating and Costing', Willy Eastern Limited, (1991)
- Ramamoorthy, M., (1987)** 'Computer Aided Design of Electrical Equipment', Affiliated **East-West** Press Private Limited, (1987)
- Rumelhart, D.E and Mc Clelland, J.L., (1987)** 'Parallel and Distributed Processing: Explorations in the microstructure of Cognition', Vol 1, Foundations, Cambridge, Mass MIT press, (1987)
- Rychener, M.D., (1985)** 'Expert Systems for Engineering Design', Expert Systems, Learned Information, Oxford, NJ 2: 30 - 44, **Vol. 2**, No 1, (1985)
- Rychener, M.D., (1988)** 'Research in Expert Systems for Engineering Design', Expert Systems for Engineering Design, ed. Rychener, M.D., **pp:** 1-33, Academic press Inc, London, (1988)
- Rich, E, and Knight, K (1991)** 'Artificial Intelligence' II ed, Tata Mc Graw-Hill pub (1991)
- Rolston, D.W, (1988)** 'Principles of Artificial Intelligence and Expert Systems Development',**Mc Graw-Hill** International Editions (1988)
- Sasikumar, M, (1992)** 'Power Supplies', Willy Eastern Limited (1992)
- Shah, M.M., (1993)** 'Design Of Electronic Circuits and Computer Aided Design<sup>1</sup>', Willy Eastern Limited (1993)
- Schildt, H., (1987)** 'Artificial Intelligence Using C',Osborne-Mc Graw-Hill, (1987)
- Simon, H.A., (1969)** 'The Science of Artificial', Cambridge, Mass: MIT press,(1969)

- Simon, H.A., (1981)** 'Cognitive Science: The Newest Science, Perspectives of Cognitive Science', Ablex pub, NJ (1981)
- Simmons, M.K and Dixon, J.R., (1986)** 'Reasoning about Quantitative methods in Engineering Design', Coupling Symbolic and Numerical Computing in Expert Systems, NH, Elsevier Science Publications B V, The Netherlands, pp 47-57, (1986)
- Smithers, T, Conkie, A, Doheny, J, Logan, B, Millington, K and Tang, M.X.,** 'Design as intelligent behavior: an AI in design research programme', Artificial Intelligence in Engineering. Vol 5 No 2, pp 78-109, (1990)
- Sriram, D and Mahar, M.L., (1986)** 'The Representation and use of Constraints in Structural Design', Applications of AI in Engineering problems, proceedings of the I International conference, Southampton University, U.K, eds. Sriram D and Aday R, A computational Mechanics publication, Vol 1, pp 355-368, (April 1986)
- Sriram, D and Fenves., (1986)** 'Knowledge-Based Approaches to structural Design', Civil Engineering Technical Report, Carnegie-Mellon University, Pittsburgh, PA 15213, (1986)
- Steier, D., (1990)** 'Creating a scientific community at the Interface between Engineering Design and AI', *AI Magazine*. pp 18-22, (Winter 1990)
- Steinberg, L., (1987)** 'Design as Refinement Plus Constraint Propagation: The VEXED Experience'. In Proceedings of the Sixth National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence, (1987)
- Sussman, G.J., (1978)** 'SLICES: At the boundary between Analysis and Synthesis', AI and pattern recognition in computer aided design, ed Latornbe, North-Holland pub co, (1978)
- Sutherland, I.E., (1963)** 'Sketchpad : A man-machine graphical communication system', Licon Labs, T R 296, (January 1963)
- Takeda, H, Tomiyama, T and Yoshikawa, H., (1990)** A Logical Formalization of Design Processes. In Intelligent CAD II, eds. H.Yoshikawa and D.Gossard, Amsterdam: North-Holland, (1990)
- Takeda, H, Veerkamp, P, J. Tomiyama, T and Yoshikawa, H., (1990)** 'Modeling Design Processes' *AI Magazine*. pp 37-48, (Winter 1990)

- Tomiyama, T and Yoshikawa, H., (1987)** Extended General Design Theory. In Design Theory for CAD- Proceedings of IFIP Working Group 5.2 Working Conference, eds. Yoshikawa, H and E.A. Warman, 95-124 Amsterdam: North-Holland, (1987)
- Turing, A.A., (1950)** 'Computing Machinery and Intelligence ', Mind 59, pp 433-460, (1950)
- Waterman, D.A., (1986)** 'A Guide to Expert systems', Addison Wesley Reading, MA, (1986)
- Warfield, J.N., (1990)** 'A Science of Generic Design', Managing complexity through Systems Design, Intersystems publications, California, (1990)
- Westberg, A.W, Grossmann, I, Talukdar, S, Prinz, F, Fenves, S et al (1990)** 'Applications of AI in design research at Carnegie-Mellon University's EDRC Artificial Intelligence in Engineering. Vol 5 No 2 pp 110-124, (1990)
- Winograd, T and Flores, F., (1986)** 'Understanding Computers and Cognition', Addison Wesley pub. co. (1986)
- Wolf, A., (1985)** 'A Text Book of Logic', (1985)
- Yokell, S., (1990)** 'A working guide to Shell and Tube Heat Exchangers', McGraw-Hill NY, (1990)
- Yoshikawa, H., (1981)** General Design Theory and a CAD System. In Man Machine Communication in CAD / CAM - Proceedings of the IFIP Working Group 5.2-5.3 Working Conference 1981 (Tokyo) eds. T. Sata and E.A. Warman, 35-53 Amsterdam: North-Holland
- Yoshikawa, H., (1983)** CAD Frame Work Guided by General Design theory. In CAD Systems Frame-Work-Proceedings of IFIP Working Group 5.2 Working Conference, eds. E M Lillehagen, 241-253 Amsterdam: North-Holland, (1983)