TOWARDS AN ELECTRONIC THESAURUS OF HINDI

THESIS SUBMITTED TO
THE UNIVERSITY OF HYDERABAD
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN APPLIED LINGUISTICS

By

Mahendra Kumar C. Pandey

CENTRE FOR APPLIED LINGUISTICS AND TRANSLATION STUDIES
UNIVERSITY OF HYDERABAD
HYDERABAD - 500 046
INDIA

FEBRUARY, 1998

CERTIFICATE

This is to certify that the thesis entitled **TOWARDS AN ELECTRONIC THESAURUS OF HINDI** submitted to the University of Hyderabad in fulfilment of the requirements for the award of the degree of **Doctor of Philosophy in Applied Linguistics** is a record of the original research work carried out by **Mr. Mahendra Kumar C. Pandey** under my supervision during the period of his stay as a Ph. D. student at the Centre for ALTS, School of Humanities. It is also certified that the present thesis or any part of it has not been submitted for any other degree or diploma to this university or any other university or institution.

(**Udaya Narayana Singh**)

Supervisor
Professor & Head
Centre for ALTS
University of Hyderabad
HYDERABAD - 500 046.

Place : Hyderabad - 500 046, INDIA

Date : February 27, 1998.

DECLARATION

I hereby declare that the thesis entitled **TOWARDS AN ELECTRONIC THESAURUS OF HINDI** submitted to the University of Hyderabad in fulfilment of the requirements for the award of the degree of **Doctor of Philosophy in Applied Linguistics** is an original research work carried out by me under the supervision of **Prof. Udaya Narayana Singh,** Professor & Head of the Centre for Applied Linguistics and Translation Studies, University of Hyderabad, Hyderabad - 500 046.

I also declare that the present work is the result of my own effort and has not been submitted to any other university or institution for award of any other degree or diploma.

(**Mahendra Kumar C. Pandey**)

Enrol. No. 92HAPH02
Centre for ALTS
University of Hyderabad
HYDERABAD - 500 046.

Place : Hyderabad - 500 064, INDIA

Date : February 27, 1998.

ENDORSEMENT

(**Prof. K.K. Ranganadhacharyulu**)

Dean, School of Humanities
University of Hyderabad
Hyderabad - 500 046.

DEAN
SCHOOL OF HUMANITIES
University of Hyderabad.
Hyderabad-500 134

(**Prof. Udaya Narayana Singh**)

Head, Centre for ALTS
University of Hyderabad
Hyderabad - 500 046.

DEDICATED TO

My

Mother and Father

CONTENTS

DIAGRAMS

CHAPTER 1.

CHAPTER 2.

CHAPTER 4

CHAPTER 5.

SCREEN DISPLAYS

TABLES

ACKNOWLEDGEMENT

It is the best opprtunity to express my profound gratitude with great pleasure to my ***supervisor***, **Prof. Udaya Narayana Singh** for his constant inspiration and encouragement that enabled me to carry out the present research work. I am indebted to him for his perspicuous guidance, encouragement, and advice during my research work. I feel that I will never be able to pay back this debt - not even through this acknowledgement.

I owe a special debt of gratitude to my ***panel members*** for their timely and scholarly advice that helped me to sail with the work to shore.

I am extremely obliged to Suchita Madam for her constant inspiration and advice that enabled me to complete the work with in the stipulated time. I am also thankful to my kid-brother and friend Bundai who was disturbed by me so much and yet who still bore with me patiently.

I am grateful to the faculty members at the Centre for ALTS, Prof. Probal Dasgupta, Dr. Panchanan Mohanty, Dr. B.R. Bapuji, Dr. G. Umamaheswar Rao, Dr. P. R. Dadegaonkar, Dr. G. Sen Gupta, Dr. K Subrahmanyam, Dr. N. Krupanandam, Dr. Sivaram Padikkal, Dr. J Prabhakar Rao, Dr. Uma Dadegaonkar and for their help and suggestions during my research work.

I am extremely thankful to Dr. Hemant Darbari and Mr. M. M. Sharma for their co-operation and suggestions regarding computer application and to Prof. Suraj Bhan Singh, Dr. Vijay Kumar Malhotra, and Dr. (Late) Ram Adhar Singh for their advice on lexicographical matters given during my research which caused me to complete the work precisely.

I am deeply indebted to my father and mother and all other family members for both moral and material support in every turn of my life. I am sure that I cannot ever free myself from their debt.

My hearty thanks are due to my friends - Dr. Shailendra Kumar Singh, Rajashree Swain, Jaya Devi Mishra, Aparupa Dasgupta, and Subrat Kalyan Pattanayak for their academic as well as moral help and co-operation.

I am very thankful to Karan Singh Negi, Sonu Subhash Chandra and Satish Shevade for their technical help and co-operation.

My heartfelt thanks are due to Dr. Vijay P. Bhatkar, Exicutive Director, C-DAC and to my colleagues at the C-DAC, Pune .

I would finally like to express my thankfulness to the non-teaching staff members of the Centre for ALTS : PVSR Murthy, A. Appa Rao, D. Mallesh, and B. Avinash for their help throughout my work.

February 28, 1998. **Mahendra Kumar C. Pandey**

TRANSLITERATION CHART

Vowels :

अ	आ	इ	ई	उ	ऊ	ए	ऐ	ऋ	ऌ
a	A	I	I	u	U	e	ai	r*	l*

ओ	औ	अं	अः
o	au	M	H

Consonants :

क्	ख्	ग्	घ्	ङ्
k	kh	g	gh	M*

च्	छ्	ज्	झ्	ञ्	(ज़्)
c	ch	j	jh	N*	(z)

ट्	ठ्	ड्	ढ्	ण्	(ड़्	ढ़)
T	Th	D	Dh	N	R	Rh

त्	थ्	द्	ध्	न्
t	th	d	dh	n

प्	फ्	ब्	भ्	म्
p	ph	b	bh	m

य्	र्	ल्	व्	श्	ष्	स्	ह्	क्ष	त्र	ज्ञ
y	r	l	v	sh	S	s	h	ksha	tra	jna

ABBREVIATIONS

CD-ROM.....Compact Disk Read Only Memory
CL.....Computational Linguistics
CPU.....Central Processing Unit
Ctrl.....Control
DBMS.....Database Management System
DMS.....Data Management System
DOS.....Disk Operating System
EGA.....Extended Graphic Adopter
ET.....Electronic Thesaurus
GC.....Grammatical Category
GUI.....Graphical User Interface
HET.....Hindi Electronic Thesaurus
HW.....HeadWord
MB.....Mega Bytes
MPD.....Manually Printed Dictionary
MRD.....Machine Readable Dictionary
MS.....MicroSoft
MT.....Machine Translation
NLP.....Natural Language Processing
ODBC.....Open DataBase Connectivity
RAM.....Random Access Memory
SD.....Semantic Domain
SL.....Source Language
SR.....Semantic Reconstruction
TL.....Target Language
VB.....Visual Basic
VGA.....Video Graphic Adopter
WFR.....Word Formation Rule

CHAPTER 1

INTRODUCTION

"Lexicography is no longer an obscure pastime performed by a small minority of introverted word collectors, but a professional activity with its own established practices and international contacts."

- Hartmann (1983 3)

1.0. PRELIMINARY REMARKS

Dictionary is a reference book which is referred to by the entire speech community for both literary and non-literary expositions. Its compactness and perfection can be gauged about only from its utilization. When a compilation is carried out by lexicographers who directly or indirectly contribute their expertise in this vital task, they usually keep the potential users of their products in mind and work accordingly. Compilation of dictionary *is* a daunting task carried out with a purpose to assign individual semantic shades and identity to each entry. A lexicographer usually follows the standard grouping criteria in order to identify and arrange semantically related words. Due to its large and complicated sorting process, a lexicographical work is perceived in two ways - either as an extremely tedious activity (and most people think this way) or as a throughly enjoyable activity (which only some highly motivated people tend to think). The obvious question that can arise here is how these two opposite reactions are handled simultaneously by the makers of these products. Obviously, the joy of building a large knowledge-base wins over or washes away the drudgery *of* reacting negatively to this goal. When after this long journey, the initial conceptualization and compilation of dictionary enter into the computational lexicographic phase, where it is assisted by a computer-system, having its own schematic designs imposed on the structure of this database, making the maximum information on individual lexical items (regarding their various grammatical and semantico-pragmatic functions) easily available to its possible end-users, the detail information that can be given on each lexical entry affects the degree of efficiency of compilation or its structure and consequently affects its implementation.

Further, the computational compiler has to take into account the basic designing and the broad framework of the lexicon himself - without depending upon whatever existing lexicon programmes he may be contemplating to use - by establishing definite parameters to be included in it as per his requirement. Moreover, the framework should confirm to some standard in the way the system is being built. Recall the statement of Nagao (1989:129) in this context who had warned long ago by saying, "One feature of a dictionary which must be given constant attention during dictionary construction is maintenance of a consistent level of quality."

1.1. AIMS AND OBJECTIVES OF STUDY

It is not an over-statement if we say that at present all aspects of life are under the influence (or, shall we say, swept under a post-structural wave) of *scienticism.* The lexicographer is not an exception, as he too leaves a long-term impact upon the lexical research by computerizing, and thus revolutionizing, the lexicon-compilation. Baker (1972:140) puts forth a philosophical view by arguing, "The modern lexicographer has been inhibited ... by the prevailing scienticism of the modern world." While advancing his suggestion regarding the solution to such inhibition, he lists out what are required : "a deeper philosophical adjustment to epistemology and to the reality of man's

conciousness - indeed to reality itself - and then, simply a deeper sociological approach to language." As all lexical items are multi-faceted, their different facets being embodiment of physical/orthographic formation, motivation - inciting action, and representation - depicting and describing modes are necessary. In this respect, the 'philosophical adjustment to epistemology', 'linguistic conciousness', and 'sociological approach' are all related to motivation, stimulating lexicographer to engage in in 'scienticism' in order to represent lexis as a different entity, complementing other facets of his creative urge. It is clear that 'scienticism' here does not affect the socio-cultural inherence underlying the lexical item(s).

The state of research in Natural Language Processing (NLP) today makes it possible to implement the theoretical advancements that have taken place in the discipline of Computational Linguistics (henceforth, CL) with special reference to Machine Translation (MT) in order to overcome the circumscriptions of manual translation. This would obviously lead to enormous speed as well as saving of time and to better utilization of energy. This will, therefore, require machine-readable electronic lexicons to be used to write a programme to mechanically transfer the written material of the SL text into a TL text. Therefore, lexicon-building today is not merely an end in itself but is intricately related to larger aims. Thus the aims of the present study are :

i. to prepare a design of thesaurus based on semantic classification,
ii. to apply computer in preparation of an electronic thesaurus for Hindi, the official language of the Indian Union, thus fulfilling an important social requirement,
iii. to utilize the thesaurus also for MT, and also
iv. to create a word-finder combining with the thesaurus function.

To fulfill the requirement of MT, spell-checking, and conformation of textual meaning, the present research work is embodied as a network prepared for Hindi Electronic Thesaurus (HET) designed to provide the users with a clue referring to the maximum information about the contexually appropriated usages of a given lexical item.

The objectives of the present research work also include highlighting the theoretical perspectives of an Electronic Thesaurus (ET) studied in terms of thematic organization of lexis with the assistance of lexical data selected from different sense groups (i.e. semantic domains), classified according to their subjects (viz. literary, scientific, and technical), and represented both semantically and alphabetically. The present research is grounded into a stipulation of existing thesauri which assign semantic classes to entries according to their own schemes but it grows out of their shadows by arriving at a scheme of categorization of world knowledge which is quite different from all other schemes. The other schemes often create problems in arriving at or deciphering contexual application of a given category or sub-category, prepared as alleviation to overcome such problems. The whole work is an attempt to reflect and take the statement of Sanjeevi (1980) to its logical conclusion. Sanjeevi (1980:126) warns that "a mere alphabetic dictionary is of no avail for easy reference to promote accurate and advanced critical studies in vocabulary."

1.2. NATURE OF LEXICON AND COMPILATION

The compilation of dictionary is based on both practical and theoretical counts. All practical applications of dictionary aim at maximizing its utility. Any practically defined dictionary is designed to provide the multi-aspectual usages to the readers, where the subject-matter is presented in a descriptive mode. A theoretically designed dictionary is principled on a network which serves to the reader with a set of specific and limited vocables. The main purpose of this type of lexicon is

to provide adequately semantically defined words used normally in the day to day activity. Obviously, the users of such lexicon in India use it as a referential tool which is well equipped in order to enable him to select the appropriate words and its adequate usage in certain specific contexts. Moreover, it also provides information regarding the structure as a part of the grammatical and orthographic information. The theoretical orientation is always changable because of the instability in the user's knowledge. Instability is an automatic process which affects the whole network of lexicon either as a part of an individual entry or the mass. It is a mental process which is normally governed by the internal - mental condition, degree of competence and performance and the external forces - the socio-cultural setting sorrounding us and by the time and space. The change as an end result in instability of language use affecting rate of addition and deletion of new words and the expansion and compression of semantic range as per the needs of communication and its conditions.

One can metaphorically claim that physical strength of a lexicon lies in the volume of entries and in their arrangement. The lexicon, of course, is compiled in a particular fashion which may be in either alphabetic or thematic order. Generally, the entries are arranged in the alphabetic mode because of their easy accessibility. For instance, the arrangement of entries in a practically oriented dictionary is done in some ordered form. But there are some types of lexicon available to the general-purpose users as their personal reference book which, by nature, have only the most more frequently used lexical items with plain semantic information. Such personal lexicon does not follow any definite order of arrangement. In the compilation of the lexicon, the compiler takes into account the grammatical and semantico-pragmatic realizations of the lexical entries. The grammatical aspect of entry refers to the parts of speech and its sub-classes as the syntactic properties and the morphological aspect refer to the construction of the world of words in terms of roots and affixes. Sometimes, the compiler gives more elaborate account of morphological processes such as inflectional, derivational and compounding depending upon the nature of dictionary. The morphological string is used for the lexicalization in actual construction. The extralinguistic features, viz. tonal break and accents, are also assigned as lexical features in the dictionary which become a part of the head word.

The accessing of semantic information on a lexis draws heavily upon the sense assignment which can be marked as synonyms. There are two processes, **comparative** and **contrastive.** which are used to analyse the semantic nature of the synonyms involved in showing the semantic componential characteristics of a lexical unit. i.e. head word. Obviously, these are rule-based processes, carried out on the basis of specific application of lexical entry. In respect of the rules used to handle the lexical analysis, Singh (1982:9) pointed out, "The lexical rules also explain the interrelationship between different lexical units in a language. The lexical rules account for the formation of new words in terms of the predictability of their acceptability or otherwise." According to him, there are three types of **acceptability** : First **actual acceptability*** defined as that of the universally as well socially accepted phenomenon being formed by well defined word formation rules; Second, **potential acceptability** referring to the fact of lexicalizational potentiality where the formation is carried out by rules which are not established in the society. Third, the **total acceptability** refers to such word formation as are "neither permissible by word formation rules nor do they have the acceptability of the society" (Singh. 1982:9).

It is quite likely that word formation rules do not exist in isolation, insulated from CON-texts. But without having an idea of the complete scenario it cannot emerge. It is also likely that they require social acceptability to balance out new forms and expressions upto some extent. The task of lexical rules is to provide information on how a word came into practice and how it can be modified

or recreated. But the social acceptability provides the practicality rather than the set up formation. Now, a question may arise regarding the degree and duration of acceptability. As far as the degree is concerned, there are some forms which can be globally accepted, whereas others are not. Likewise. in case of duration some are accepted for a long span of time whereas others only for a certain period. Such phenomenon leads into the following set of principles:

1. Follow the most common word formation rules (WFR) which can be easily accepted among the speech community.
2. Try to avoid vagueness in sense expression so that it makes the speaker comfortable in comprehending the expressed reference.
3. Rules should be clearly exposed.
4. Nature of the rules can be viewed in terms of application levels.
5. The changing in the original form after the application of word formation rules should be easily marked out.
6. Rules should be applied in such way that they cover the expressive modes like metaphorism as a part of the connotative and stylistic backup.
7. The situation and context be clear and vivid.
8. The future utility should be presupposed by which the speaker gets an idea about stability and survival of the lexical item(s).

Dictionary as a book deals with four sets of activities, viz. identification of words at large, detailing each of them as an individual entity in terms of diction, explaining each one as per one's requirements, and providing with orthographic information and significations with some explanatory information such as phrases and idioms. Because of their purpose and signification, classification of dictionary-types is a very crucial aspect in any lexicographical work. Similarly, the purpose of compilation can also be very dynamic. The whole task can be divided into two classes : **pre-processing** which includes pre-planning, material (data) collection, selection, gradation, representation and **post-processing** which includes rectification of the printed draft. These processes depend upon nature and scope of dictionary one plans to prepare.

Though dictionaries are meant to serve different utilities, their format and subject may also differ from one type to another. There are many parameters involved in attesting to varify and determine the class of a dictionary. They are as follows :

1. **Areal axis** : whether selected lexical items belong to particular region of the given speech community or are spread over the entire geo-space.
2. **Social enitity** : Whether entries are bound by particular socio-cultural entity or they are general.
3. **Range of subject-matter** : Whether it covers all subjects or some limited ones.
4. **Entry specification** : Whether the selected lexical items are general or special covering specific domains.
5. **Nature of entries** : Whether it covers specifically the slangs, archaism or is general.
6. **Mode of entry set up** : Whether they are arranged according to the alphabetic order or whether they follow a semantic order.
7. **Representation of entry information:** whether it is encyclopaedic or normal.
8. **Language coverage** : whether the dictionary is monolingual or bilingual or tri/multilingual.
9. **Period** : whether the dictionary is diachronic or synchronic.
10. **Scope/Purpose** . Whether it will be used as a subjective or a normative reference.
11. **Users** : Whether the users are general or special.

12. **Logical view:** The lexical representation is done in haphazard fashion or in some systematic and logical manner.

As far as limits of compilation is concerned, overlapping naturally occupies the centre space. But in the case of encyclopaedic dictionary such overlapping has no scope or has a very rare chance because it covers the information about the well known incidents and personal details in the case of persons or characters. Though there are some clarifications that lie in the overlapping of lexical items, it leads the user to get information in its appropriate place rather than search for it at its place or through cross references. Such problems can be sorted out to some extent in the current computational compilation methods. It should be pointed out that the overlapping may be of two kinds : lexical overlapping and the thematic overlapping. The previous deals with the repetition of the lexical items which occurs at the rate of how many contexts it covers and with how many times it occurs as synonyms of other lexical items. Like the previous one the latter also deals with subject-wise contexts. Moreover, it also marks the repetition of the theme in general.

The analysis of any entry finds many characteristics bound by the vision of scope and coverage of the dictionary maker, with a definite focus on some peripheral marks on socio-cultural notes and linguistic perspectives. On this ground, the dictionary comprises lexical information pertaining to the linguistic aspects, viz. grammatical and semantic information, as associative individual treatements, whereas in the case of encyclopaedic coverage, it includes personal details like names of persons, places, and literary works alongwith the human fact of knowledge which is extensively treated in the individual record. The encyclopaedia, unlike lexical/general dictionary, is mainly concerned with the extra-linguistic information with limited aspects but it is usually very brief information without any detail. Aspects of human knowledge include well-known characters, environmental bodies like names of plant, and animals and the historical events. It is noted that sometimes the socially and culturally bound lexical items cannot be explained with a single word. Rather they require some socio-cultural notes. The lexicographer treates these notes in the precise and unambigous manner which enables the user to understand with a single glance. Moreover, to determine the concepts underlying a lexical item one can take recourse to the definition form. Finally, the polysemous word having many semantic shades should be given in the respective branch of subject from where they belong.

The diachronic dictionary deals with words across time whereas the syncronic ones with a particular point of time. Though it is very difficult to recognize the mark of a period in synchronic form on any given lexis, the lexicographers apply some criteria in order to draw the periodic line in the whole span. In fact, there are many criteria and principles which come into analysis and compilation of the dictionaries according to their vary nature and purpose, and they can be accordingly classified under the heads of special dictionary and the general dictionary.

All lexicographical studies draw heavily on linguistic approaches which define the lexicographic concepts and help in collecting the necessary information to handle or undertake all issues in a proper fashion. Lexicography is an applied wing under the broad discipline of linguistics. Therefore the linguist becomes the resource for methodological as well as the semantic principles provided to the lexical study of a language. The main linguistic roles in the lexicographic context are recapitulated through the lexical study. It requires the principles as well as strategic interpretation as per the exigency of the formalism that is chosen. The lexical ***formalism*** pays a specific attention towards the language study in the specific mode which is described in respect of its applications. The nature of the linguistic applicability - a theoretical potentiality of the lexicographic study - is crucially noted in the disciplinary system of formalism. This applicability may be circumferenced by

the contextual parameters. The contextuality is elucidated according to the semantico-pragmatic reality. That is why, the context, by nature, is destined as consequences of the semantico-pragmatic space providing situation, intention, and expression. The fundamental linguistic theory makes certain theoretical assumptions which are marked as the providential contrivance of the language situation and necessary for application of it in the existing time and space. It also provides a convenient and rule-bound method which covers the exceptionality in language-usage to a great extent.

The lexicographic premises are motivated through the application of linguistics in both cases, language specific as well as universal. In the case of language specific application, it is concerned with the establishment of phenomenon which is demarcated in the same direction of the existing theoretical intimation, whereas, in the later case, the phenomenon is found almost in all languages with the same or similar typicality. Hence, the lexicographer has to take into account these features in order to compare the linguistic peculiarity in the whole attestation. The lexicographic work must obviously be carried out by referring to phonetics, phonology, morphology, syntax, semantics and pragmatics. These accounts are either concerned directly or treated indirectly to frame the lexicographic network. Such work is crucially dependent on the coverage or the availability of adequate background research.

Lexicographic work will include the compilation or making of the dictionary or its different types, viz. learner's dictionary, practical dictionary, comprehensive dictionary, and so on and so forth, and the different modules of compilation, namely thesaurus and encyclopaedia. It is noted that these classifications are based on the entry selection, coverage, and representational formalism. To take an example, the learners dictionary is compiled with certain practical considerations which enforce the following points :

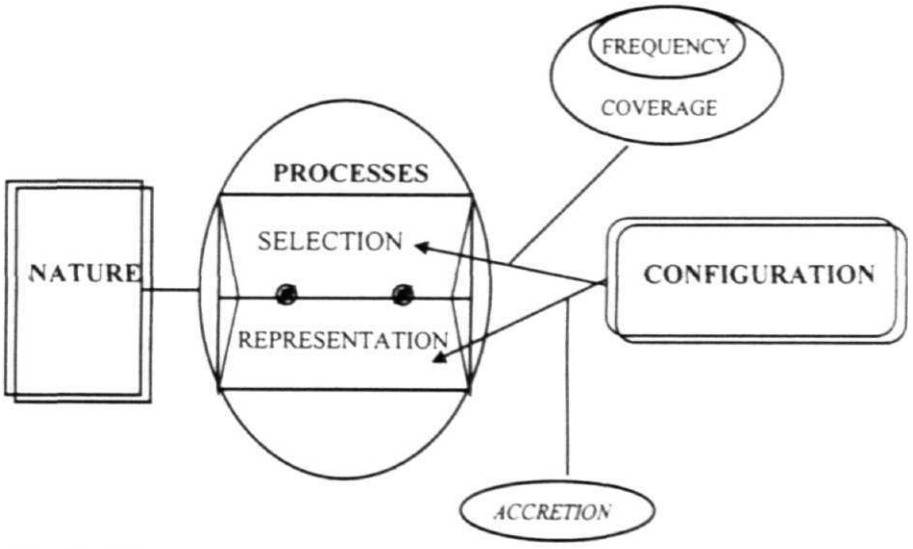


Diagram 1 : Inter-dependency between Nature and Processes of Lexicon-making and Determination of its Configuration.

Any lexicographic work requires a sharp attention and it is particularly true in the public domain general-purpose dictionary or thesaurus-making. The lexicographer has to take many crucial decisions in the compilation process depending upon the nature of the work. He has to give an account of the word selection procedures which enable him to come to the aimed model coverage. He has to pass through two sub-processes, viz. ***selection*** which is governed by the features of

acceptance, alternation and decision on the words to be included in the thesaurus or dictionary. The second sub-process of compilation is *representation* which consists of mode, module, and explication. These issues are discussed in detail in Chapter 2.

At this juncture, it seems to be the right time to look into the nature of lexicographic compilations vis-a-vis different types of dictionaries. Moreover, the aims and objectives underlying in their framing are also highlighted in terms of their subject coverage. Here a table could be presented to show the classes of the dictionary compilation aimed at the different perspectives which are very briefly described :

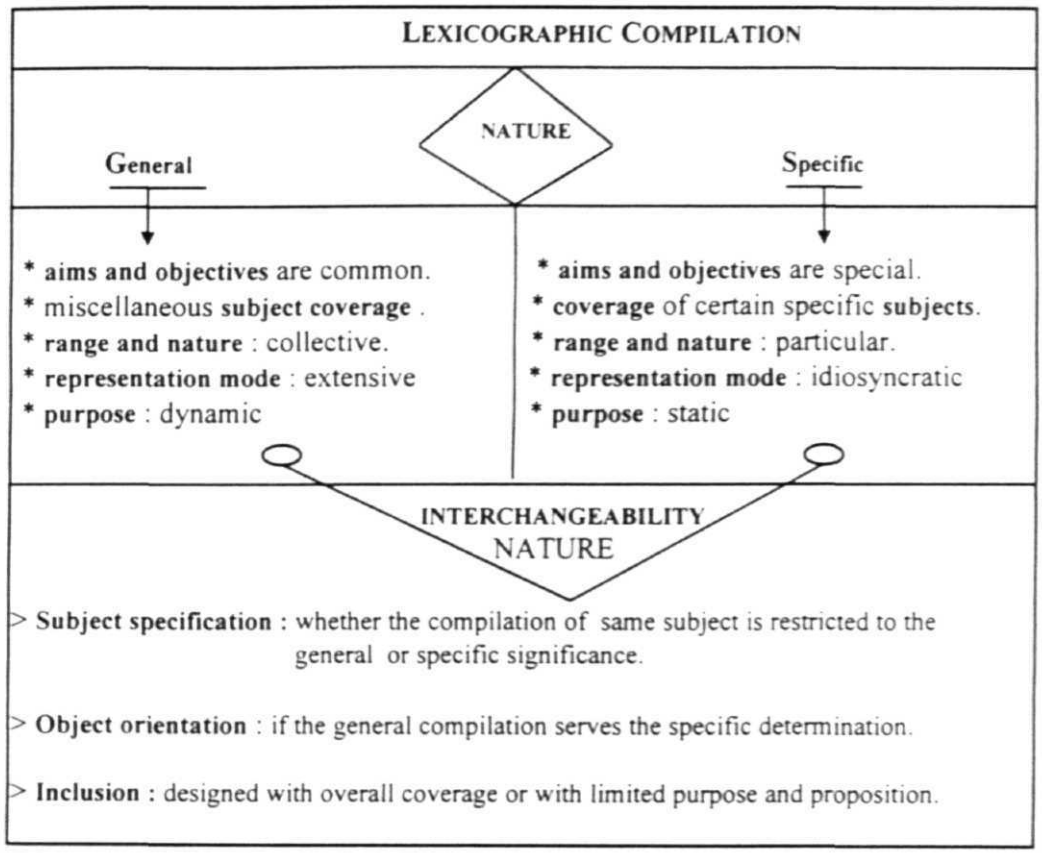


Diagram 2 : The Distinction between General and Specific Compilation while Preparing a Lexicon and the Areas of Interchangeabiliry.

The distinction drawn between general compilation and specific compilation is worth noting in the context of lexicographic work because it provides a theoretical outlook for design and embodiment. Let me take some types of dictionaries alongwith their characteristics and purposes as instances. The encyclopaedic dictionary is very common in its nature but specific in its presentation. By nature, it covers the linguistic and extralinguistic features of language. It embodies both encyclopaedic information, viz. the historical documents about the known incidents which are taken as entries in the encyclopedia. To quote Singh (1982:13) will be appropriate here because he says. "The encyclopedia are more concerned with the concepts and objects of extra-linguistic world, that is, the things and in a narrow sense they may be called "thing book"...Their aim is to present information, as noted earlier, on all aspects of human knowledge. The items presented are more of denotational character including names of plants, animals, diseases. They also give historical events, geographical features, biographical sketches of important personalities."

The above definition provides a brief information about the nature of subjects included in an encyclopaedic dictionary. Though the present issue is to highlight the diction-oriented design, the information included into a general encyclopaedia becomes the resource of an encyclopaedic dictionary. Such dictionaries are a combination of an encyclopaedia and a linguistic dictionary both in terms of extent of information provided and style of presentation.

The important issue that can be raised here has to do with the distinction between an encyclopaedic dictionary and a linguistic dictionary. Sometimes drawing such distinction becomes difficult as they show overlapping because the features of linguistic dictionary which deal with the lexical items require some kind of encyclopaedic information any way, and vice versa. With this reality in mind, one can claim that they are associative in subject representation, except that the degree of range may make them vary. This claim is, however, weak in its argument. But it is still true that the information handled in an encyclopaedic dictionary is more comprehensive and fact-based where as the information given in the linguistic dictionary may be called a mere factual citation. The distinction has to do with not only the explanation but also the subject matter. The encyclopaedic dictionary covers the extralinguistic knowledge-base which may be treated in linguistic terms. But in addition, it may also be possible that the linguistic phenomenon is explained in terms of extralinguistic features and space.

A linguistic dictionary presents the common features of lexical items and is compact in respect of its arrangement. Its represenation is compact as it usually has a precise design. If one looks at the classification of the dictionary critically, the following points of classification will emerge as instances differentiating different traditions of dictionaries :

Set 1. Subject-oriented classification (AlMS-based FORMALISM)
- a. linguistic dictionary vs. encyclopaedic dictionary
- b. general dictionary vs. special dictionary

Set 2. Time-oriented classification (PERIODIC BOUNDEDNESS)
- a. historical dictionary vs. etymological dictionary
- b. synchronic dictionary vs. diachronic dictionary

Set 3. Contact-oriented classification (QUANTUM OF LANGUAGE)
- a. monolingual dictionary
- b. bilingual dictionary
- c. trilingual dictionary
- d. multilingual dictionary

Set 4. Language-oriented classification (LANGUAGE COVERAGE)

- a. *intra-lingual dictionary*
  - (i) standard language dictionary
  - (ii) dialectal dictionary
  - (iii) sociolectal dictionary
  - (iv) gender-based dictionary
  - (v) register-specific dictionary
  - (vi) acronyms/synonyms, etc. dictionary

b. ***inter-lingual dictionary***

(i) general-purpose bilingual dictionary
(ii) translator's dictionary
(iii) technical dictionary

Set 5. Direction-oriented classification (MODE of REPRESENTATION)

a. reverse dictionary
b. forward dictionary

Set 6. WFR-oriented classification (MODE of SEGMENTATION)

a. roots dictionary
b. affixal dictionary
c. dictionary of bound forms
d. compounds dictionaries
e. phrasal dictionaries

The classificatory system followed in grouping dictionaries, as shown above may not cover ail the names of dictionaries but covers almost all types of dictionaries. Critically the following possible motives can be exerted from this classificatory system :

1. The set 3 and set l.b are common to all sets.
2. The set 5 is uncommon.
3. A contrastive approach has been followed in classification.
4. Emphasis has been laid on the angle of marking, e.g. the dictionaries in which language is being emphasied will be classified as under Sets 3 and 4.

The aims of compilation of the dictionary governs all these perspective strategies. For instance, whether one is compiling a monolingual dictionary or a bi-tri- multilingual dictionary either in the reverse or in the forward modes as synchronic or diachronic in its general or special formalism. Now a question can be raised regarding the efficacy of dictionary, viz. whether or not it is possible to include all subjects or approaches into one dictionary? The answer must be obvious. It should never be possible because the lexical study in terms of applicability and acceptabilry is beyond the individual knowledge and it might create complexity rather than simplicity which is the fundamental purpose of creating a dictionary.

1J. THESAURUS : *DYNAMIC PERSPECTIVES*

The thesaurus has its own style of compilation with an account of pattern *of* semantic relationship of words in their textual-linguistic as well as contextual behaviour. Katre (1965:8) rightly gives his opinion about the nature of lexicographic work by stating. "Modern Linguistics has not sufficiently realized: Lexicography is both a science and an art. However, its place as a science may better be discussed under the more general term. Lexicology, while as an art it may be considered under its general name of Lexicography." Here, one can notice that the term "lexicography* is used in two different contexts having hierarchical relationship where the first 'lexicography' (occurs in the statement : 'lexicography is both a science and an art") is used in broader sense in its hyperonymic use. i.e. 'lexicography* assigns only in context of as an art'. That is why, to elucidate such a quandary, one can advocate and suggest that lexicography and

lexicology are two different terms standing for related fields but with different focus. These subjects are, of course, internally related.

1.3. 1. ON **DEFINING THESAURUS**

Any attempt at defining a term itself shows how difficult it is to decipher its characteristics once the product is available in the market without much theoretical discussion, etc. The word 'thesaurus' (Engligh < Latin < Greek word 'thesaurus') means "treasure". The existing definitions of thesaurus given in the different dictionaries are imperfect in consolidating the total-sum of concept being imported. Let us examine some of them critically:

1. "storehouse of information, esp. dictionary or encyclopaedia."

2.i.a. "a collection of concepts or words arranged according to sense."
b. "US (American Use) a book of synonyms and antonyms."

2.ii. "a dictionary or encyclopaedia."

3. "a repository of words or knowledge; hence, a lexicon encyclopaedia."

4.i. "a book of selected words or concepts, such as specialized vocabuiaary for music, medicine, or the like."

ii. "a book of systematically classified synonyms and antonyms."

Neither any one of these definitions given above is explicit and adequate in comprising the integral conceptual information, nor do they cover the multi-dimensional aspects of thesaurus except one or two aspects. The critical as well as conceptual inadequacies of these definitions could be noted as follows: (1) They do not accentuate overall countenances of thesaurus. (2) Some *of* them do not distinguish between the terms, viz. thesaurus, dictionary, and encyclopaedia. (3) In some dictionaries, more than one definition are used which, in fact create problems for the readers or users. Hence, the definitions can be viewed as very controversial in their nature but can still be accepted as definition of synonyms (cf nos. 1, 2.ii. and 3). The rest of these definitions are at best partially true. For the present to ensure the imperfection found in defining the thesaurus can be put aside, one can redefine it as "a treasury book of world-knowledge dealing with the comprehensive concepts and information of words having been included and arranged either alphabetically or semantically; where the semantic classification can be done according to subjects and sense and sub-sense assignments of lexemes" (Pandey. 1995).

1 J.2. NATURE OF LEXICAL WORK : *ART OR SCIENCE* ?

Compilation of the thesaurus carries both descriptive as well as prescriptive orientations. Thesaurus as an art can be surmised at the level of drafting of the model which has to be applied in different phases of designing and then in the actual building up of the product. The initial phase will depend on the planning as it exists in the compiler's mind. Here, he should be able to undersign the pre-planned network cognitively. According to McGregor (1985:123), "Dictionaries are like engineering feats. They must first be conceived and then written: first designed and then constructed." Hence, not only 'dictionary' passes through two phases but also a theasurus. The

term 'design', in fact, is abstract or cognitive in its nature. The artifact of thesaurus compilation is an outcome of the compiler's thought and memory. It is undertaken in the process of selection and gradation.

The later phase of building a thesaurus has to deal with the translation of the cognitive design on paper into an output performance including all information as per requirement to complete proposed/aimed lexicon. Here, the topics taken into account are an anatomy of overall network of lexicon providing the physical appearance and arrangement of entries in substantial mode as intended to be set up. The second is a vehicle' for the first one. This part is mainly concerned with representation.

Thesaurus as a scientific activity mainly relates with the linguistic perspectives related to the logistics, or the representation occuring at the mechanism of designating its grammatical category, representing its morphology -- in case of the compound lexeme and at the sub-entry formation stage, and setting up of the citation/examples. Besides, it takes into account the socio-cultural aspects as per requirement. The whole process is thus handled systematically and scientifically. In addition, an electronic compilation looks up the computer corpus and application. The mechanism involved in the preparation of an electronic thesaurus can be classified under two heads : linguistic and non-linguistic perspectives vis-a-vis the electronic enterprises. The former is a language-oriented avocation where the compiler takes care of linguistic methodology - rules, principles, and procedures, by delineating the linguistic facts and separating them out from the non-linguistic information, whereas, the latter is exercised only when the compiler takes an assistance of a computer in facilitating the compilation of thesauri or other lexicons. Consider the following diagrammatic representation of these distinctions:

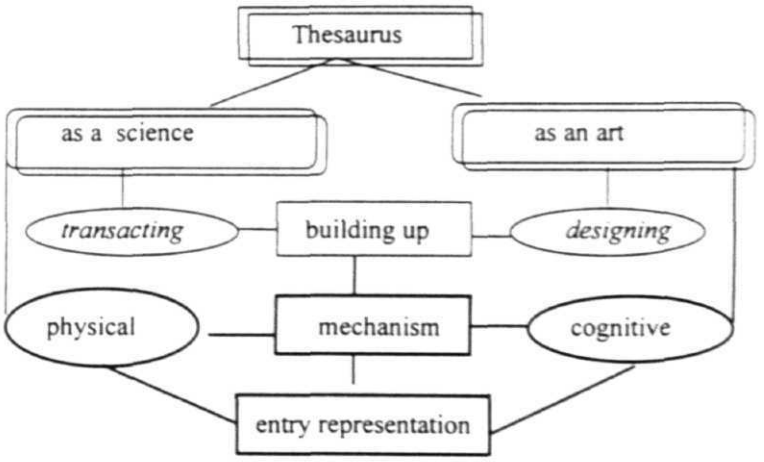


Diagram 3 : A Classification Showing a Thesaurus as an Art and as a Science.

1.*3.3.* TYPES : *ALPHABETIC AND SEMANTIC*

As far as the number of languages are concerned, a thesaurus is usually compiled monolingually. The classification of thesaurus is ingrained in its schematic content comprising the module-format framed according to the modes of representation of lexical entries and their meanings. In this regard, in fact, the thesaurus can be contrieved into two frames due to their variation in arrangement of lexical entries where one may have alphabetically arranged entries and

another may be based on semantically classified arrangement of entries. Although being different in nature, they share the same subject having synonymous and antonymous representation. In fact, they have their own approaches. Let us highlight some of them.

1.3.3.1. THE ALPHABETIC THESAURUS

The alphabetic thesaurus is framed consistently with entries arranged alphabetically (depending upon the standard arrangement or order of letters in the source language or following the English alphabetic order- especially if the language concerned has adopted the English script to a large extent), e.g. the theasurus of synonyms and antonyms. Any entry can be searched out and spotted here at a single place, i.e. at its respective alphabetic order-cum-place. The compiler assumes that the readers or users know the sense-implication and assignment of the given lexical item once they are discriminated for their context. It has a homogenous representation. The crucial fallacy of alphabetic thesaurus is that there is no way to define a lexical item, except only by giving synonyms. But sometimes, especially in Indian languages, there are such mono-words which do not have any synonyms but which can nevertheless be defined. These can be either ignored or taken into account in order to just to decode it by adopting misleading synonyms with intuition of rationalization of sense-representation.

13J.2. THE SEMANTIC THESAURUS

The semantic thesaurus is formulated according to subject-matter and their sense-domains. Here, an entry can be easily searched out through the devives of index and/or cross-reference. The compiler assumes that the readers do not know much about the entry and its sense-assignments being searched. But he would like to elucidate about that respective entry and its context-bound-semantic usages with some annotation. Unlike, the nature of homogenity. which is obviously possible to find at times in the semantic correlationship among the entries, it is noticed here that especially in an electronic thesaurus-model the alphabetic arragement can be safely ignored as it is taken care of by the programmes anyway. But it occurs at two levels : at the inter-contextual level, the arrangement carried out a subject-domain search as in literary, scientific, technical, and legal contexts, whereas at the intra-contextual level, especially in the subject-wise classified entries, for instance, the Hindi words, **janma** 'birth' and **youth** 'adult' belong to a particular domain in our scheme of things, i.e. SELF, and therefore, they will be grouped under **SELF** domain in their respective place.

1.4. LEXICOGRAPHIC COMPILATION : *LINGUISTIC APPLICATION*

Linguistics in its theoretical as well as the applied areas accommodates lexicographic work. Through the application of linguistics to dictionary-making, the lexicographer sorts out the lexical problem precisely. In what follows, a brief description of the areas in linguistics and their contribution to lexicographic compilation is given.

1.4.1. THEORETICAL AND APPLIED LINGUISTIC SOURCES

A brief account of areas and approaches in linguistics and their utility for lexicographical work is given below :

Theoretical

1. Phonetics	:	transcription
2. Phonology	:	its account
3. Morphology	:	GC, WFR and Derivation, etc.
4. Syntax	:	syntactic functions/locus, case trapping
5. Semantics	:	meanings, etc.
6. Pragmatics	:	contextual usages and constraints

Applied

1. Sociolinguistics	:	social accounts
2. Psycholinguistics	:	thematic intention
3. Ethnolinguistics	:	cultural information, etc.
4. Historical Linguistics	:	historical development, etc.
5. Computational Linguistics	:	computer applications
6. Statistical Linguistics	:	statistical accounts of language

The linguistic approaches to the lexicographic compilation enables the lexicographer as well as lexicologists to identify and decipher the basic problems with lexical items. With the help of phonetics and phonology, the compiler can give an account of pronunciation and its variation caused due to the regional influences. These linguistic sub-disciplines also account for the flexible resource of the rule and gives the procedures of phonemic interpretation of pronunciation and also gives a descriptive representation of pronunciation variations. The morphology plays a vital role in determining the lexical items in terms of their root form, base form, derivations, and the inter-connection between and among words - especially in compounding, abbreviations, and acronymization. The lexicographers and the lexicologists must have morphological interpretation insights to undertake the description of an item or in hypothesizing about the internal composition of lexical items. The configuration of a lexical item is accessed through its morphological structuration. The exposition of derivational forms of a lexical entry is analysed by certain morphological rules called WFRs. The nature and apportionment of affixation processes involved in the derivatives are marked and classified according to the functional as well as positional approaches. It also gives an account of the base and its roots. Whether a lexical item stems from a particular root, or whether it is a polysemous item, etc. are tackled by the etymological explication. Like morphology, the syntax also provides a clue to understanding the position and function of the lexical items. The semantics and pragmatics also enable the compiler to sort out the problems of synonyms or equivalents. The compiler can then decide about the contextual usage of a particular lexical item.

The disciplines under applied linguistics are equally useful in the compilation of dictionaries. A historical or an etymological dictionary provides the clues for the historical development and etymological interpretation of lexical items. An equally valuable information on entries come from applied areas such as sociolinguistics. Similarly, psycholinguistics provides the necessary extralinguistic information related to the intensity of the lexical items which are crucial in their contextual analysis. Ethnolinguistics also gives the details about the variation in use fron one ethinic group to another within a speech community. The computational linguist assists the compiler to decide how the computer can be helpful in compiling a dictionary. The statistical linguist also enables the lexicographer in terms of the frequency count and other statistical information on words.

1.4.2. COMPUTATIONAL LINGUISTICS: *AN ELECTRONIC STRATEGY*

The present era has been a 'witness'to revolution in computer-production and application in different realms - language, science, business, and technical corpus with congruous developments - speech synthesizer, morphological analyzer, and data management system and programmes handled with the help of softwares designed for machine language-codes. To name some important computer applications that deal with human language, one has to depend on the new discipline which emerged as a result of collaboration between 'Artificial Intelligence" and 'Linguistics', and which is called Computational Linguistics. Computational Linguistics is a new technical discipline which emerged only in the recent decades as a part of Applied Linguistics with an agenda to fulfill the requirement of effective inter-communication through the machine as is required in the world context. This kind of development is required in the exchange of valuable information within a short spectrum of time. As expected, application of the computer has created a revolution in the Linguistic Study in general, too.

Natural Language Processing or NLP is the sub-discipline of Applied Linguistics mainly related to the application of the computer in the study of language. The basic aims are to make it possible to use natural languages to interact with the system as well as use the computer system to analyse or synthesize natural speech. Under this sub-discipline some branches of Linguistics, both theoretical as well as applied, are studied. The nature of application of the computer and the practical utility of it depend on the the nature of work. For instance, in dictionary building, the lexicographer takes the help of computer in feeding/storing the data and in manipulating it to see if word generators are in order or to think of possible neologistic patterns. All these data are then used for the preparation of the electronic dictionary and thesaurus. The advances in programming make it possible to handle the related work exhaustively as intended by the linguist/programmer. The positive motif behind establishing this technical application is due to utilizing the system to get maximum benefit and to get the work done precisely. Moreover, it helps to save time and energy which are the crucial factors among the human resources. Obviously, to come across the problem of time and energy, one has to make use of some technical aids; in such cases, the linguist takes the assistance of computer to fulfill the goal. Moreover, the work carried out with the assistance *of* computer is expected to meet the required precision and perfection in nature. The present work is obviously an instance of using the advancements in NLP in lexicographical work.

1.5. ORGANIZATION : *CHAPTER-WISE REVIEW*

The research carried out in this thesis can be divided into theoretical perspectives of thesaurus, viz. the lexicological details as well as practical aspects of lexicography including computer applications the highlight of it being an innovative semantic classification which underlies the design of this programme. The thesis is divided into five chapters which are as follows:

The first chapter entitled **INTRODUCTION** incorporates the preliminary current remarks on the lexicographic traditions - from the ancient Indian to the electronic, the motive being to underscore the importance of such research assignment. The critical highlight of it is the definition of thesaurus, its types - alphabetic and semantic, its nature - both as an art and as a science, and a discussion on the computational linguistics and NLP along with its different perspectives. There is a brief account of the thesaurus in general along with the alphabetic and semantic types. The present work is chiefly a semantically classified thesaurus, and therefore, the main focus is on this theme. The alphabetic thesaurus does not require such attention as it is more like a list of synonyms.

The second chapter is titled **THESAURUS : *ATTRIBUTES AND FORMATION*** dealing with the multi-dynamic nature of words as seen in terms of time, space, and region. A brief account of the attributes, categories, approaches followed in compilation, drafting system and devices show its diffrent dimensions. The form-sense relationship and the critical observation given by some experts are highlighted here with some propositions. This chapter provides a comprehensive exposition on dictionary-making in general and thesaurus-compilation, in particular. It covers the extended information of the previous chapter, i.e. introduction, where a brief literature was made available on the complications involved in this kind of work. Here, the semantic deconstruction and reconstruction are highlighted as lexical semantic theory. Moreover, this chapter gives an account of the computational linguistics in respect of its application in preparing an electronic thesaurus. It also highlights the theoretical perspectives of the computational linguistics. The nature and range of application of computer in the linguistic research, especially in the lexicographic compilation, is also considered in detail. It also comprises a description of the linguistic problems and their solutions through the artificial intelligence methodology. The knowledge-based application in language is considered here as the exploration of linguistic entity and logical ideation. The lexico-semantic approach to the lexicographic and lexicologic determination are discussed in brief. The idiosyncratic properties of the lexical items are also highlighted.

Moreover, the second chapter also deals with the attributes of the thesaurus in general and electronic thesaurus in specific. This issue is given at the rate of theoretical features which are required to determine the thesaurus-network. The attributive features of electronic thesaurus are highlighted in terms of the entry arrangement by following certain well- defined strategies and parameters. Those parameters and strategies are discussed as methodological and theoretical set-up for the proposed electronic model for Hindi thesaurus. The grammatical categories, meanings, derivation, spelling variation and so on and so forth are supplied in detail. This also deals with the methodological formalism of creating an electronic thesaurus. Here, the existing thesauri are scanned in terms of their characteristics - entry formation, application features, infrastructures, goals. and coverage range. Moreover, the nature of thesaurus is pointed out with the citation of some well-known models of thesaurus. In terms of framework, it is divided into two classes : oriental and western models. For oriental, we have selected models right from Sanskrit language to modern Indian languages, whereas, in the case of western model we have selected from English language with different infrastructures.

The chapter three, namely. **SEMANTIC DOMIAN-ASSORTMENT** presents the classification of the world knowledge which is divided into eight broad classes. There is a brief discussion on the semantics and associative meanings or on the environments of each semantic class and sub-class, but there is not much detail given here, since the categories and classes are discussed here without examples. The semantic domains as given in the chapter comprehensively are claimed as typologically valid to the extent that they are applicable at least for the South Asian languages, but these can also be claimed to be fairly general, if not universal. Here eight broad classes are also shown alongwith their sub-classes. The sub-classes are further divided into sub-subclasses and this goes on in a hierarchical manner. The sub-classification of its higher class or sub-classes are determined logically and as per requirement. The entire network is of course constrained by the current state of knowledge and native wisdom.

The chapter four which is entitled **ON-LINE NETWORK** is a computer programming oriented chapter dealing with the application of computer in the preparation of Graphical User Interface (GUI) which makes the actual intercommunication between the user and the computer possible. For our purposes here. Visual Basic 5.0 has been selected to prepare the GUI and to

interlink the database in order to display the input as output fashion. The whole chapter outlines two important aspects of the thesis. The first is related to the introduction of the package. The introduction of Visual Basic 5.0 is shown through the its run-time screen display in order to show the utility of the package in the preparation of an electronic thesaurus. In this context, the theoretical aspect is concerned only by providing the points rather than giving a detail information because the aim is not to give an account for the package but to show the applicability of the package in thesaurus preparation. The second aspect is concerned with the actual design created here based on the GUI. There are six screen displays alongwith the thirteen information screens which are highligted. The properties of these screen displays are used to show different aspects of the entries in this chapter. The functions of those screen displays are discussed in a precised way. Moreover, the interlink among the screen displays are also given briefly.

Charpter four also shows the thesaurus as an on-line network viewed from an electronic theorem point of view. It displays almost all recorded input into output alongwith processing and implementation. In this technical chapter, the attributes of present modules are given. The data base model and the query system and its range are discussed here in precise terms. The algorithm of the data base design is also discussed in this chapter. An introduction to the computational lexicography and computer as an electronic machine. The application of computer in lexicon preparation, especially thesaurus-building, is discussed in detail. The function of computer as a compiling tool, or for programing implementation, and the accessibility and capability of the programme are also dealt with here. As a theoretical chapter, it deals with the computational power in thesaurus compilation. The solutions of the complexity in the compilation of thesaurus are highlighted. Moreover, it shows the nesting of the different attributes as formatted in the database form. It also introduces the Microsoft Access database system. It also contains some screen displays as introduction to the Microsoft Access 2.0.

The fifth and the last chapter entitled **CONCLUSIONS** includes the concluding remarks regarding utilization of computer in compiling an electronic thesaurus. The limitations and gains in all such thesaurus-creation endeavours are also noted. It also consists of what is the present and future utility of proposed model of thesaurus for different purposes. Moreover, the future of computational linguistics, especially of computational lexicography, are also discussed in brief. Moreover, it also highlights the application of the computer in natural language processing to create user-friendly tools to generate mass-use. It also gives a brief account of the limitation of the current model and its reasons.

1.6. SUMMARY

The lexicographic work in India has a very old tradition which has been existing right from the Vedic period to the present times with different profiles - models and varieties. The compilation of lexicon - a dictionary or thesaurus was always considered essential here because *of* usefulness *of* this tool for language-understanding. The output of lexicon depends upon the subject matter, availability of source, range, and the readers. With the development of language, it is quite difficult to select all possible existing words and their manipulation but one can, at this stage, go for a fair sample of words of a standard language. Such practical problems faced by a compiler can also be sorted out with an electronic device now. Therefore, the compilers must make use of the computer to bring in uniformity in compilation of lexical entries with minimum effort devoted to get the maximum output. The computer as a compiling, contextualizing and cross-referral aid enables the dictionary-maker to optimize his or her time in compiling a lexicon which can perform this a multi-dynamic function.

CHAPTER 2

THESAURUS : *ATTRIBUTES AND FORMATION*

"The lexicon has come to occupy an increasingly central place in a variety of current linguistic theories, and it is equally important to work in natural language processing. "

-**Levin** (1991:205).

2.0. PRELIMINARIES

The current era demands exhaustive use of computers in language analysis and application in the fields of speech recognition and synthesis, morhophological analysis, sentence parsing, semantics as well as in discourse analysis, and lexicon preparation. In the case of lexicon-building, a large part of which requires mechanical handling of a huge database, the computer helps in various ways explained below. The computer is an electronic machine used to compute some complex vocations related to computation and other solutions which are manualy difficult to carry on and are time consuming. All given tasks can be solved within a very short time. Broadly, in terms of functioning, computers can be identified with two aspects: hardware and software. The hardware broadly refers to three parts : (1) The keyboard consisting of various defined keys used to type characters, numbers, and mathematical signs and having certain function keys such Tab. Cap Lock, Shift, Ctrl (=Control) and F(unction) keys etc. (2) the central processing unit (CPU) which is a store-box that packs in the hardware mechanism consists of electronic mechanic devices used to process the information loaded in through the help of keyboard, and (3) the monitor which as a mirror displays the visualized information fed into the CPU through the keyboard. The software is a packege used to mould, categorize, arrange and represent the data depending on what one wants to do with it. The data feeding and its use depend upon the nature of software as well as dictate as to what kind of software is required. The software is made differently in order to fulfil a user's requirement of developing packages. The input to output process is tackled by the software. It containing *electronically-sensitized* storage cells comprising machine's 'memory' called locations or registers.

2.1. COMPUTATIONAL PREMISES

The use of computer in compiling lexicon, especially thesaurus, is very practical, worthwhile, constructive, feasible, and productive in terms of "automation' in this building process which might be semi- or full/complete in nature. The rational objective behind the "computer-use' is to get the maximum output based on an input with least substantial efforts. The task given to computer to manouver the data at hand is performed by it automatically which enables user to manipulate it more precisely and scientifically in comparison to manual efforts needed to do so. The practical benefit of computer use to compile a lexicon or thesaurus is that it can help at both stages of data management and its manipulation.

2.1.1. COMPUTATIONAL LINGUISTIC ALLIANCE

The conceptual contrivance of MT gives congenital birth to electronic thesaurus or lexicon. The electronic thesaurus, as a sub-module of MT, is used to handle textual analysis and synthesis. Though it is a challenging task of shifting from numeric premise to alphabetic symbols or signs, but at the current stage of development of computation, they become a reality. A textual analysis requires recognition and representation of texts as a knowledge based system. Regarding the application of computer in language study, one is constrained by linguistic rules. Therefore, different scholars have

given different view-points of computer assisted language study. Knowles (1983:183) had highlighted three scholars' views, as given below, about the application of computer in language study:

Garvin (1962) **__*. Demarcation of three modes of computational linguistics** :

1. language data-collection,
2. computerised applications of linguistic research results, and
3. automation of linguistic research processes.

Lamb (1961) _^ **Overture of a five-fold sub-divison** :

1. a computerised production of 'aids' to linguistic - and also literary- analysis,
2. language 'automation' - both analysis and synthesis,
3. the simulation of language dynamics,
4. information retrieval, and
5. the statistics of linguistic phenomena.

Martin (1974) —• **Four ways of computer use in literary and linguistic applications** :

1. as a classificaton machine,
2. as a calculating machine,
3. as a control/checking machine, and
4. as a simulation machine.

Let us recall what Batori *et al* (1989:XVI) had pointed out in this respect: "A theoretically oriented definition of the location of Computational Linguistics as related to linguistics would start out from the importatnt role of data processing in the research field and specify the interdisciplinary links between language and the computer from a linguistic angle." The critical elucidation regarding the location of Computational Linguistics, as shown in the above statement related to the data processing, and elucidated to the restricted sphere of language processing as carried out in textual processing. The computational linguistic research has bi-directional conceptions of *language-orientedness* and *computer-aid as* interdisciplinary and concerted application of Artificial Intelligence and Linguistcs. As far as 'language-oriented research' is concerned, its accessibilty covers different areas of language depending upon linguistic procedures and principles. It is a natural language-oriented knowledge-based practice. In the case of computer-aided research, application of computer in the linguistic analysis assumes centrality, where the *linguistic phenomena* are accessed and processed computationally depending upon the efficacy and capability of the system used. Hence, the notion of *natural* become *artificial* here. The pragmatics of transition from the 'natural' to the 'artificial' is still an evolving phenomenon.

The computational linguistics is born as an outcome of the co-relation between Applied Linguistics, the application of computers to linguistics, and Applied Artificial Intelligence, where the problem in language-oriented research is resolved. Batori *et al* (1989:XVI-XVII) observes the innovative perspective in Computational Linguistics, when he says. "The autonomy of Computational Linguistics can be justified by innovations that are assigned to four levels of knowledge." These levels of knowledge are as follows :

1. **The new methods** : Adoption of new and different philological models rather than using the usual ones in order to solve the problems of language data-processing and text-processing.

2. **The new insight**: The problem of the data-processing or text-processing can be handled with appropriate amount of rectification which opens up a new perspective of computer-aided research.

3. **The new type of problems** : The advance computational and technological applicability to language research gives birth to the problem of degree of application, i.e. full or partial, in terms of automation in processing and accessing the expected output.

4. A **new structuring of the field** : The language analysis in terms of data-processing requires a comprehensive scheme of field set-up with appropriate infrastructure, unlike the conventional linguistic field set-up.

These innovative views of Batori *et al* show the methodology of novelity. Each application will be off-beat as compared with the previous module alongwith a precise problem solving strategy. The novel model had more efficiency when compared with its proto-formalism. According to this position, the cognitive science is also a part and parcel of Computational Linguistics, when Batori *et al* say, "So Cognitive Science and Computational Linguistics are not rivals or mutually exclusive conceptions. The accentuated interdisciplinary, especially the extending of the horizon towards psychology and the study of volitional aspects of language (ranging beyond pure information conveying) are also attractive perspectives for Computational Linguistics, especially in the area of language simulation." (XVII). Now, in the current standing, the computer is used as a fast, capable and efficient tool for accessing and solving problems of respective disciplines.

2.1.2. DYNAMIC PERSISTENCE : *COMPUTATIONAL LINGUISTICS*

In the previous section, we have seen different strategies proposed heuristically in respect of Computational Linguistics. Computational Linguistics, as a sub-discipline of Linguistics touches upon or covers almost all areas of the theoretical linguistics. The views of the experts, as cited above, is more or less as the basis in defining its specific area within linguistics. In the ongoing stream of research in the hi-tech electronic system, the axiom for solving the natural language becomes the ultimate mission of the computational linguist. Hence, any interrogation into the artificial intelligence which touches upon the nature of language and cognitive capability contribute significantly in developing this field. In fact, although computational Linguistics is a relatively new discipline, it is growing quite expeditiously. But, still the question of disciplinary coverage is open - whether the computer will be the substitution or adjunction to the natural profile of human dictum. It is quite obvious that by birth and sourrounding, the deficiency is determined as an inheritance. Therefore, computer as a man-made and man-governed object will always be under the control and constrain *of* human cognition. But, it does not mean that in terms of capability, it will be lower than the man in all respects. In certain kinds of job, it is superior to man. For instance, in the degree and time spent on accessing data from a huge part of information, where human capability' fails to challenge the machine.

The computer, as a tool of assistances, is used in both Theoretical Linguistics and Applied Linguistics. A number of major areas of Linguistics, viz. syntax, lexicography, translation, and morphology, have benefitted from computational intervention. The computer is used in all these areas as a tool implemented for complete as well as partial automation, as per one's requirement. Recall the appropriate statement of Grishman (1986:6) in this context who gives his conception of Computational Linguistics and Theoretical Linguistics in the following words: "Although both are ultimately concerned with understanding linguistic processes, computational and theoretical linguists have rather different approaches and outlooks. Computational linguists have been concerned with developing procedures for handling a useful range of natural language input. They are (in general) willing to accept approximate solutions ... understanding of the entire process of natural language comprehension and generation." But still they are internally dependent upon the shades of each other. Grishman (1986:6-7) further states, "Theoretical linguists, in contrast, have focused primarily on one aspect of language performance, grammatical competence - how people come to accept some sentences as grammatical and regard others as ungrammatical. They are concerned with language universals - principles of grammar which apply to all natural languages - and are interested in finding the simplest, computationally most restricted theory of grammar which can account for natural language. They hope thereby to gain some insights into the innate language mechanism which enable people to learn and use languages so readily. In their efforts to evaluate alternative theories, they are often led to study peculiar sentences which some computational linguists would regard as pathological."

The *linguistic argumentation* that underlies in this position on the natural language analysis being presented here is a conjecture which works as a methodological strategy. Practice of exploration of language in terms of its development is based on periodic endurance having time bound flow right from the origin to current development and application. Such penetration aims at identifying the language characteristics as a chronical semantic and orthographic development. This type of linguistic attitude towards exploratory research gives an account of the presuppositon of the linguistic set-up in terms of *naturality* or a kind of distinctive linguistic orientation. Language, as a consequence of *continuum statrum* is considered as a conspicuous ideational process: where words as individual strings of a whole linguistic entity mould and are moulded in a come-and-go process in the following contexts :

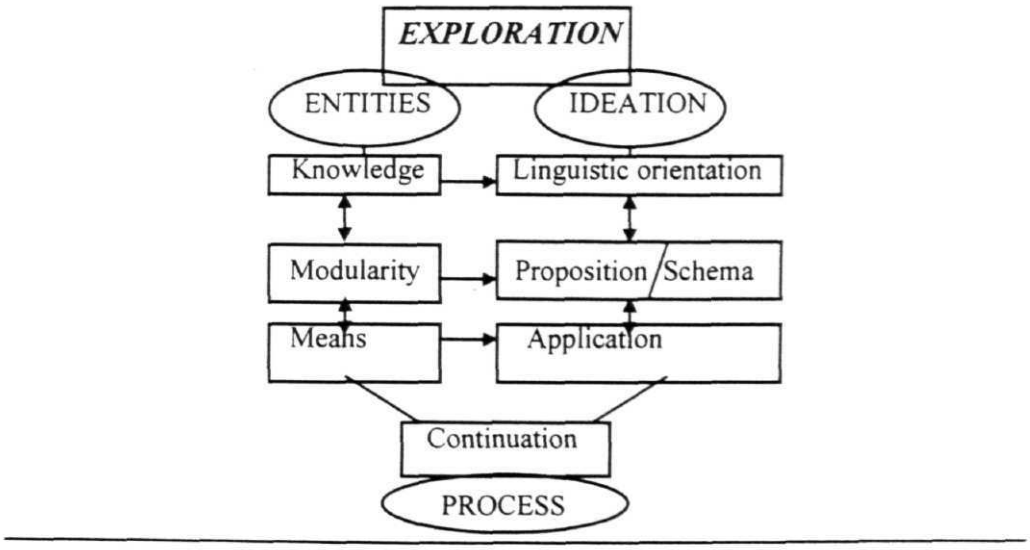


Diagram 4 : Linguistic Exploration and Language as an Ideational Process.

The language, as a spheric entity carries sensible discernment under some kind of shape and shade which is used as a *means* of communication. The process of language development, in terms of sense assignment, modularity, and means, is a continuous innovation. The continuity in pragmatization can be motived by past experience depending upon the linguistic changes - in orthographic shape and semantico-pragmatic shade. The development in a specific environment and context justifies its synchronized explanation as a part of dichronical perspective; where this synchronized explanation is marked as an innovating idea having string-like-standpoint in a periodic fashion. In order to set-up any kind of thesaurus, especially a semantically classified thesaurus, one can follow well defined theorem having the capability and potentiality of giving an account of synchronized explanation in terms of semantico-pragmatic shades. Indeed, it requires scientific and logical thinking over a rational assesement of it. The evaluation of linguistic fact carried over in terms of the historical perspective of language development and application in synchronized strings is a factual decision as shown in the following diagram :

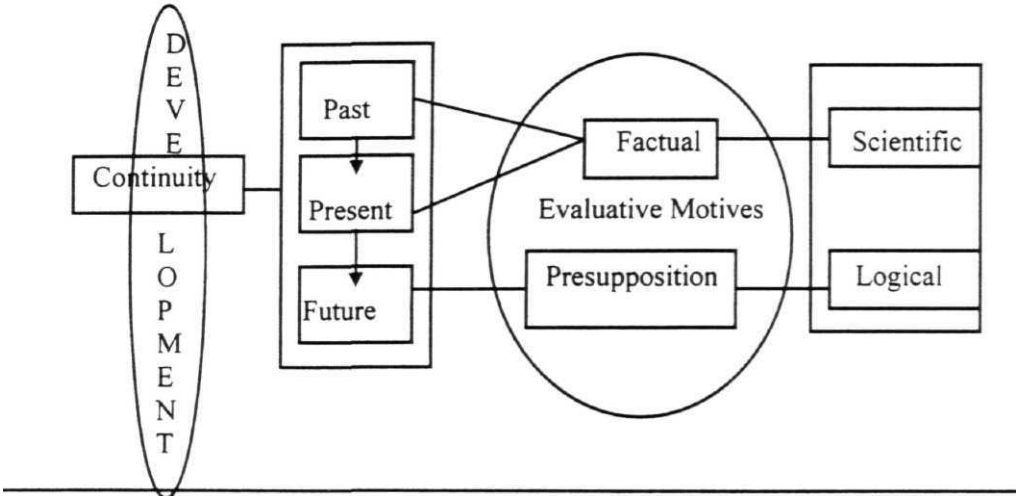


Diagram 5 : The Development of Words in different Periods as a Continuous Process and their Evaluative Motives.

The significance of language as a part of a universal schema nested with the communicative process and the artificial language with its own philosopy and reasoning, we assume that the idea of a thesaurus was conceived within the framework of a certain linguistic paradigm but soon this paradigm had stopped being really influential. Once this type of word collection was a common thing in the world, and it went on attracting the attention of people because it could serve some of their practical needs. The artificial language is a knowledge-based phenomenon. The mobilization inspires the experts to attain the peak of language application through automation-aids within worldwide space. *Universality* of language is still a question open to challenge - whether the efficacy of artificial being (i.e. through an electronic machine) can be used to process and access human language to cover its pragmatization of application. Still the goal is quite far to reach. Of course, the natural language has its own vitality as contrasted with the artifial language which has only restricted application at present.

It is universally true that each and every construct is bound by its own limitation. It is not surprising, therefore, that natural language has its own limitations. The knowledge in any one of them affects the semantico-pragmatic judgement of linguistic application due to the vary nature of a native speaker's innateness and also because of context. To comprehend the sense and its context, one should have world knowledge. Hence, world knowledge does not refer to the orthographic shape and

to very little knowledge of application, but it refers to the comprehensive application with all possible senses assigned in different contexts. So, the word becomes the heart of the language.

Though computer is 'heartless', it can nevertheless 'understand' language having its own register and encoding system. It keeps the interaction with the human being through the codes. That is why it is an interpretator and a creator, too. The computer becomes more capable in processing the language for lower level applications. But it may be possible that it can go nearer the human understanding by encreasing its capability as noted in Garvin (1966:XI) statement, "The first-named characteristics of computers can be called their data-processing capability, the second, their logical capability. The data-processing function and the logical function can be used as the two poles of a continuum on which the nature of linguistic problems and their solution with the aid of computing equipment can be discussed."

2.2. LEXICO-SEMANTIC ALLOCATION

The semantically classified thesaurus is by nature based on the semantic network, especially on the lexical semantic standpoints. This is a device to look into the application of the word usage in the specific contextual set-up. It also enables the user to recognise the semantic quintessence of the contextuality. Recall Pandey's (1994:163) statement in this context, "The nature of semantic perspective in terms of lexicographic work, i.e. compiling dictionary, is mainly pertained to the two-fold-relationship between lexical item and its possible meaning(s) at horizontal level and between one meaning to another meaning at vertical level." Though it is very difficult to handle the lexical semantic in the thesaurus, but in the computational application such problem can be sorted out with serious computing effort. Regarding to the complexity, Hays (1967:17) says, " Linguists have to deal with more complicated materials, and deserve better tools than simple index registers for keeping track of them."

2.2.2. PERIMETERS OF WORD

The multidynamic nature of a word which is our product of the region where that word origined, time referring to the particular period when it was coined, and space referring to the development of the word up-to-date and its place amoung other similar and related words, explain these variations and account for the interrelationship which helps in assigning meaning to it. The nature of a word is determined by its usages and contexts which give insights proffering prodigious profile of semantic assignment. The nature of a word can be categorized in terms of following : semantic range, positional nature, referential nature, and functional nature. They are comprehensively highlightened in the following passages. Hays (1967:174) says, "A dictionary can contain an encoded semantical analysis of individual items. For instance, this can take the form of a hierarchical classification of meaning, somewhat like the organised listing in Rogefs thesaurus." Though his view regarding to adjacent appraisement and solutions related to the semantic application to dictionary preparation, which is more appropriate to that of thesaurus.

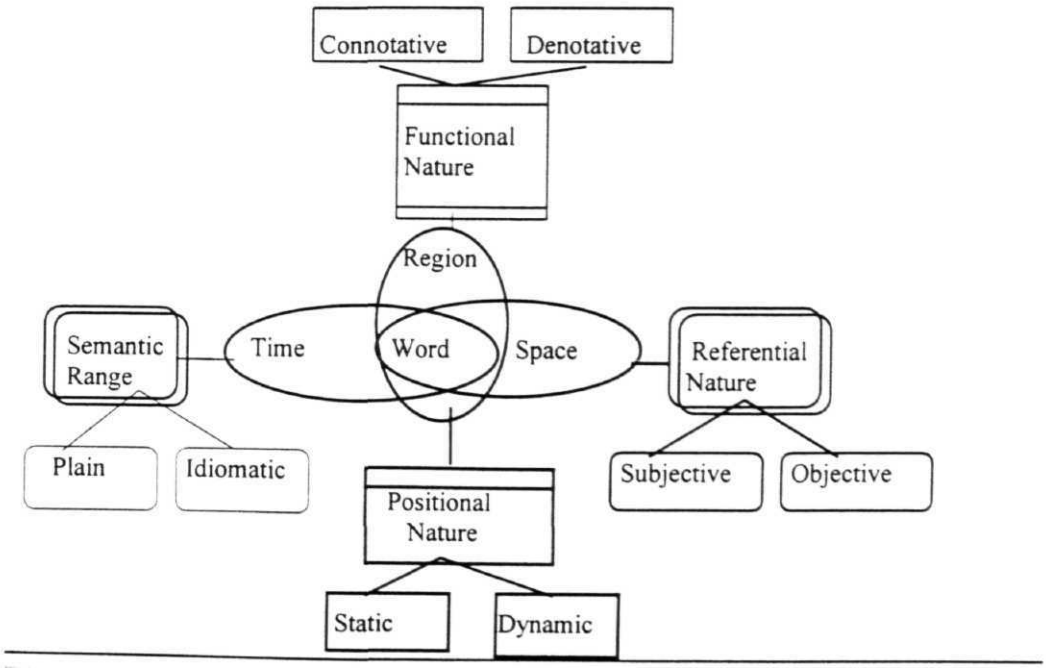
2.2.3. MULTEDYNAMISM AND IDIOSYNCRATICISM

Word conveys certain references by implying the sense-representation associated with its semantic implications. In fact, the nature of a word, in terms of semantic implicational coverage, can be considered as plain or direct and idiomatic. The idiomatic here refers to the suggestive meaning of a word (having many semantic assignments), whereas the plain refers to the direct or primary meaning of a word. This distinction mainly depends upon the semantic coverege related to the possible context(s) in which a word can be used. If any word occurs in only one context it will be considered as plain semantic nature of word whereas in the case of idiomatic nature of a word it

occurs in more than one context. Moreover, the meaning assigned by plain mode of expression is always static in its nature in all situations/environments. In contrast, the meaning assigned through the idomatic mode of expression varies from situation to situation.

The other quality of word is its positional nature which can be viewed in terms of two different perspectives — static and dynamic. The static nature of a word mainly deals with the position which is always firm in most contexts without affecting or changing the meaning which is merely one in number. But the dynamic nature of word refers to the multifacets of word which is changable from one environment to another. In other words, the static nature refers to unchangablility of word in its original semantic expression whereas the dynamic nature of word is related to the use of that word in different situations alongwith assigning different contextually/situationally suitable meanings.

A word mainly refers to either any concrete aspect of meaning related to the object or ideation or to any flexible subjective ideation which mainly arbritrary in thought. The very nature of referential aspects of word is due to the nature of the referent to which word refers. If the referent is more concrete, unique, and objective the referential information conveyed by a word will be objective in nature. In the case of subjective referential information, the nature of referent to which references are made by the word, makes it more subjective. This flexible semantic scope is related to the thought and ideation. The word functions differently in conveying its meaning. If the expression is idiomatic in nature it will be said that the word connotates the meaning whereas if the word is used to refer to the direct object without any idiomatic expression it will be said that the word denotes the object. The word functions as a connotative agent to convey the secondary or suggestive or supplimentary meaning while it functions as a dennotative agent to refer to the primary or first meaning. The senerio of peripherical aspects of word in its semantic assignment-context as discussed above can be presented in the following diagram given below:



Digram 6 : Representation of the Peripheral Aspects of Word and their Interrelationship.

2.2.4. LEXICALIZATION: *RECONSTRUCTION AND DECONSTRUCTION*

The general lexicographical practice followed in this monolingual electronic Hindi thesaurus is aimed to provide from a word to its many meanings and vice versa mapping system. The semantic density of the lexical item covers the thematic reference oriented in the contextual sphere. Such dynamic mapping is carried out within the language boundary according to its possible usages. The Lexical Functional approach followed here provides necessary theoretical information to the practical problems faced by the lexicographer when he/she compiles a lexicon. The different modules of lexicon compilation undergoes certain lexicographical parametric channels in order to identify the associative componential strings. The network of an electronic thesaurus framed with its own enchancement gives the automatic resolutuion of the retrievals. The module proposed, as part of research, is mainly based on the following theorem :

- Word, as an expression-media-unit, has potential power to express senses according to its capability determined as in contextual domain. There are two processes involved in the meaning determination. They are 'semantic reconstruction' and "semantic deconstruction".

The semantic reconstruction (SR) is a processes where the meaning of a construct can be determined as an innovation involving the combination of the features of two senses of the same word or other words. SR process can be *intra-componential* or *inter-componential.* In the intra-componential process under the semantic reconstruction, the features of the two senses, assigned by the same word, comes together in order to create its other senses. In the face of inter-componetial analysis somehow the distant processes ocurring between two senses assigned by two different words are important, where by combining the features of these two senses, the new sense is created which assigns one of them as the appropriate sense. Consider the following illustration :

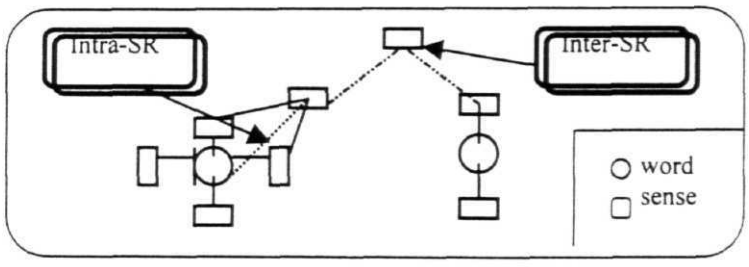


Diagram 7 : Semantic Reconstruction

The semantic deconstruction (SD) deals with the processes of *weak* and *lost* senses when a word has more than one sense and due to certain course of the reasons, either it loses one of them or becomes weak in its usage. This process can be shown in the following diagram.

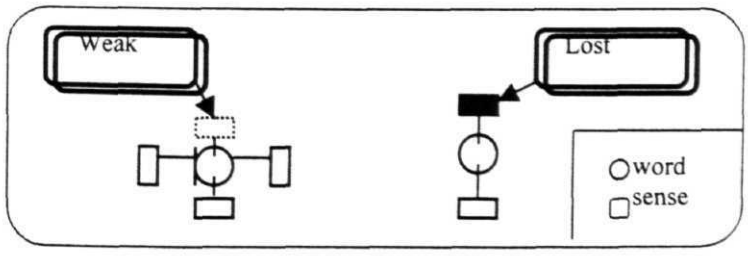


Diagram 8 : Semantic Deconstruction

- The semantic network of a lexical item is nested according to the determination of the context. Indeed, the linguistic approach towards language analysis, depending upon its lexical usage, is determined according to the speech community's conceptual points of view.

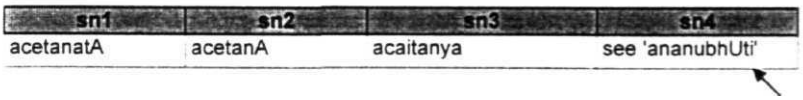
- The inter-disciplinary linguistic approach is another way of viewing the interrelationship between linguistic notions/concepts. According to Zgusta (1971:18), "the mutual dependence (or interdependence) of "lexicon" and "grammar"... is of first class importance".

2.3. UNDERSIGNED MECHANISM

The meaning of a word is determined by the *extraction process*, dealing with the range of a given word in usage. If it occurs at the lexical level application of the task of meaning extraction is carried out at that level. But when the meaning concerns either sentential or discoursal level aplication, the meaning of that word can be extracted at the respective level. The extraction process will make the meaning representation easier, but such process should be clear in the principles - giving unambigious clear-cut specific contextual sense. Recall Singh (1982:84) who says, "A meaning is tentatively fixed for the word from the first extract. It is later on verified to its appropriate meanings taken out of all the possible contexts." Indeed, the practical process becomes more difficult when meaning assigning becomes dynamic. In such case, the lexicographer becomes handicapped in showing the complete process of extraction of that meaning in the given context, but as a clarification he will give some clues as reference either as cross-reference or in bracket within the entry itself. Here two methodologies are adopted to show the cross reference. They are:

- Click twice in the Listbox of the screen display form 6 (see chapter 4) which will copy in the Headword editbox of the same form.

- If the synonyms are a part of the same headword and if it exceeds the range of data fields defined in the database, the cross-reference, as shown by the arrow, in the scheme presented below :



The mode of expression itself conveys certain explicit meanings of a word. In the completion of the speech cycle - from speaker to the hearer - there are so many factors governing the expression and its reference. Normally, those factors can be classified into three components, viz. speech orientation, transmission and perception. The speech orientation comprises the speaker's attitude, behaviour, and physical condition. In the case of transmission of speech, the external sorrounding governs the utterance. The perception as endpart of the speech cycle, is governed by the meaning determination and interpretation.

2.4. SENSE-EFFECT-DISCRIMINATION

This is the right place to highlight the issue of defining meaning which has long been trapped under the unending debate on what is the meaning of meaning ? Although to some extent, it is difficult to define meaning in a perfectly acceptable manner due to the variation and critical controversy underlaying various views expressed by each individual scholar. The interpretation of 'meaning' as a part of language, in terms of its formation and usage, could have an ideosyncratic view based on the general perception of *object.* The *subjective positivism* and *logical interpretation,* on the one hand, and *objectivism* with *empirical interpretation* on the other hand, may lead one to take the naturalistic standpoint of language.

The mapping of meaning of a word is done by the thematic reference conveyed by it in a specific orientation. The mapping system itself is a dynamic phenomenon which follows the lexical functional approach comprises the theoretical information related to the word and its relationship as well as usage with the extralinguistic sphere in order to express intended viewpoints. The reference conveyed through the word assigning the senses is determined by certain processes and principles involved in the meaning expression where the semantic application of a word is marked according to the range of application. The linguistic as well extralinguistic application of language analysis (including meaning analysis) depends upon the nature of social composition of a given speech community. The case of the inter-disciplinary linguistic approach, as stated earlier, magnifies the whole utterance according to the linguistic components. Each language has its own structure which is well constructed and well principled in terms of application as well as formation. It should be noted that there is no direct relationship between word and sense assignment, but in between, there is some object- orientation which helps in designated recapiulation.

The componential analysis is one of the major approaches used to analyse the lexical units having specific features. The identification of semantic features of a set/group of the lexical items which are marked componentially, carry some common features which may be called the global features. The important issue looked into in this research is that each language system has its own system of construction and diction application. The lexical units are arranged according to the language specific contexts-demands. Of course, the language carries some common features, i.e. is called the universal ones. In a very rare case one lexical item has 'absolute' synonyms due to 'flexibility' of lexical usage. Though they have some similiarity in the semantic appilication at the macro level but at the micro level they even differ form one another due to the context specific application. The context specific application of a lexical item is sensitive' where the application can be restricted to the net tip of the semantic locus. The task of the lexicographer is multi-modal because he has to find out all the applications of the lexicon being compiled. In the case of thesaurus, he normally follows some strategies, as in the earlier chapters, as they are the synonymous representation and the semantic classification . The previous one deals with the entry representation as in the listing-of-synonyms mode whereas the latter one with the arrangement of the lexical item according to the the subject and semantic domains. From the utility point of view, the listing of synonyms without giving any semantically covert information creates a problem to the reader in finding out the exact contexually appropriate words. But in the case of the semantically classified representation it makes the reader's identification of the problem very easier.

2.5. THESAURUS-QUIRK

Thesaurus, as a reference book, contains words arranged either alphabetically or semantically conveying the information about the orthographies and sense-references. They are meant to meet the specific needs depending upon the its nature, user's knowledge, and their points of view. The nature of dictionary determines the coverage, utilization and the range of users. For instance, the scientific dictionary will be restricted to the coverage of scientific terms, but not literature of other disciplines. Dictionaries framed in the descriptive mode with a large amount of data gives the frequently accepted meaning rather than giving the detail information about the rare semantic application. But such information requires interpretation of the changes it undergoes and acceptability. One may recall the statement of Gajendragadkar (1973:l3-14)~about the descriptive nature of dictionary here : "Open-ended sets which are not delimited by means of a rule and which deserve enumeration are included here. A descriptive dictionary in order to be truly useful to all types of needs arising from different points of view, should not only utilise the dimensions of form and meaning, the associated collocation, peculiar syntactic situations but also exploit the parameters of dialectal variations and registral differences arising from profession, function, caste etc. It is obvious that in order to be useful, the word entries are to be supported by suitable quotations and references.

bringing out, in full, the distribution of the word in its various dimensions. The number and the type of lexical items such a dictionary will include will depend upon the aim of the compilation and the likely clientele it proposes to cater it. Words which are stablized in the speech community are given in their accepted form and meaning."

2.5.1. ATTRIBUTIVE FEATURES

The arrangement of entries and its various uses, as a part of the dictionary-crafting, depends upon the nature of the dictioanry. For example, in the case of encyclopaedic dictionary, the entry contains the personal notes of writings, personal vita of important persons in the field and so on and so forth. There are two types of the meanings, primary and secondary, which are widely accepted. There is still controversy among the scholars about how to decide about them. Obviously, if it is the question of sense assignment, the context will occupy the central place. Neither the context nor the word itself can lead to any consequence of determination. But, taking decision over the determination of primary and the secondary meaning is based on the nature and sense-assigning range of word. For instance, if it is a mono-semantic word, the sense will be primary. And if it is a poly-semantic (polysemous) word, one sense will primary and the rest will be secondary, depending upon the nature of assigning ranges - lexical, sentential, and discoursal. Regarding the sense and its determination. Kelkar (1973:63-64) says, "The critical test is whether the meaning is the one which will most readily suggest itself to the user if the vocable is *mentioned* out of the context rather than *used* within a context. In relation to a synchronic account of a language that is no longer current, one may want to recast this test slightly : the basic meaning is the one that an editor interpreting and annotating a text is most disposed to pick up so long as there are no clear contexual pointers in a different direction. In the rare cases where a vocable may have more than one equally viable neutral senses....".

Grammatical information needed to assign in any kind of lexicon can be defined as the 'morpho-syntactic peculiarities' of lexical unit designated, in terms of parts-of-speech. gender, and number, according to its fuctions in the context. There is operational interrelationship between grammar and lexicon in such way as Jackson (1985) states out. "More precisely, a Grammar describes the syntactic arrangemens of classes of items; it (Dictionary) describes the kinds of grammatical "meanings" (e.g. plurality, tense) that may be realized in a language, and the formal means (e.g. inflectional endings) by which those meanings are realized." The remarkable point is that the thesaurus cedes only "grammatical category", but not "described grammar".

The definition of 'definition' in a dictionay is to give a concrete stability of concept being explained in verbatim execution with the help of minimum well-set required words having utilitarian acceptability in conveying the sense-assignment. It is quite obvious that defining a word in 'concrete stability' covering overall excellent nuances of its sense-implication assigned in all contexts is only to some extent possible due to what Bolinger (1985:69) advocates. "Undamaged definition is impossible because we know our words not as individual bits but as parts of what Pawley and Syder (1983) call lexicalized sentence stems, hundreds of thousands of them, conveniently memorized to repeat -- and adopt - as the occasion arises. And also as part of an associative network involving words of similar and opposite meaning, words of similar sound, similar spelling, similar register, and affect." It is also quite true that nobody can drop down his pen until one defines a word to determine its optimal significance enclosing almost all. to the possible extent, contextual semantic references.

The lexical meaning, here, just stands for the synonyms (single lexical unit) of a particular headword. The lexical meaning may be referential, adoptive, denotative, or connotative. Recall the factual and logical arguement of Laird (1990:x) here who says. '"A thesaurus ... can help by providing words that are not very close synonyms but do provide the writer with a better way of

saying what he wants to, especially by using concrete words, names for real things." It is widely accepted that form and meaning is coherently interrelated. It means without form there is no meaning. In case of lexicon, besides simply semantic theoretical application, this 'form-meaning relation' can be advocated as well as judged at two-folded dimensions. The first deals with 'head-entry" functioning as form and its semantic shades functioning as conventional expressions and conveyed through other signs (words) which are considered as meanings of it. Therefore, they (headword and its semantic shades) are entitled under the concept of 'form-meaning relationship". Likewise, the meanings (synonyms) of a particular head-word, encoded by some physical properties (orthographs), function as forms because they have taken some physical conventional signs and then the head-word in each case itself functions as meaning. Such relationship one can advocate either within the same entry where the head-word functions as the main entry or in the respective place of its semantic shade where that head-word comes under meaning sphere as semantic shade of its meaning. The practical use of this format helps the user to get more comprehensive information about the usages of lexical item taken into search.

The compilation of thesaurus or other kinds of lexicon passes through some channel namely selection, gradation, and representation. These processes are mainly based on the aims and objectives of underprepared lexicon where the orientational dimensions, viz. the range of lexis, network of representation, aims of compilation, involved language(s) with its structure, stipulation, and limitations, and the readers are taken into savoir-faire verifiably. The process of 'selection' is carried out at two levels : manual and computational. The previous having with multifacets ensues at the echelons of subject-domains, and lexical availability alongwith their grammatical function, and semantic implication in all situations. Such work is mainly conducted only with the staff of pen and paper. The latter includes an account of computer application utilized in terms of selection of software, and network. Thesaurus, like other lexicons, is an open ended wordbook consisting of hundreds of thousands lexis alongwith other required information, viz. grammatical information. The compiler has to select maximum possible lexical items and their semantic shades having multidynamic dimension such as dennotative, connotative. figurative, referential, and objective. Here, in this research work, the selection has been done with restricted data containing the lexemes belonging to the different subject-domains and senses, and sub-senses used commonly and frequently among Hindi speakers. The subject-classes that are not taken into contemplation in selecting words here include neologisms, proper name, archaic words, pure Sanskritised words, dialectal words, and lone or borrowed words.

Gradation with higher step of process of compilation sorts out the material unworthy of compilation. Hence, the vocation of grading of the material being used for compilation is processed to finalise it. The selected material (lexemes and their meaning-shades) is difficult to compile in the lexicon. Therefore, one has to possibly make a choice about those lexemes and thier semantic shades being included. The graded material is represented in the mode of pre-well-defined assorts. The present research precisely intakes the representation comprising the feeding of data (i.e. adequately selected and graded lexical items alongwith their meanings and grammatical information) for promised output. The sense representation can be classified either in the individual mode or globally. In the case of individual mode, the senses of particular lexical entry and in the globle mode, the subject-domains are taken into account.

Preparation of thesaurus has to settle some practical problems of compilation such as variation in spelling, ordering of grammatical category, numberization of semantic assignments, nature of cross-reference, and indexing system. The peculiar feature of language is varying in spelling but having the same meaning(s). Such variation is found due the variety in pronunciation of words of a particular language spoken by a segment of the speech community. Like other languages, Hindi has also some variations. For example, y and j are alternants as in the Hindi

saying what he wants to, especially by using concrete words, names for real things." It is widely accepted that form and meaning is coherently interrelated. It means without form there is no meaning. In case of lexicon, besides simply semantic theoretical application, this 'form-meaning relation' can be advocated as well as judged at two-folded dimensions. The first deals with 'head-entry" functioning as form and its semantic shades functioning as conventional expressions and conveyed through other signs (words) which are considered as meanings of it. Therefore, they (headword and its semantic shades) are entitled under the concept of 'form-meaning relationship". Likewise, the meanings (synonyms) of a particular head-word, encoded by some physical properties (orthographs), function as forms because they have taken some physical conventional signs and then the head-word in each case itself functions as meaning. Such relationship one can advocate either within the same entry where the head-word functions as the main entry or in the respective place of its semantic shade where that head-word comes under meaning sphere as semantic shade of its meaning. The practical use of this format helps the user to get more comprehensive information about the usages of lexical item taken into search.

The compilation of thesaurus or other kinds of lexicon passes through some channel namely selection, gradation, and representation. These processes are mainly based on the aims and objectives of underprepared lexicon where the orientational dimensions, viz. the range of lexis, network of representation, aims of compilation, involved language(s) with its structure, stipulation, and limitations, and the readers are taken into savoir-faire verifiably. The process of 'selection' is carried out at two levels : manual and computational. The previous having with multifacets ensues at the echelons of subject-domains, and lexical availability alongwith their grammatical function, and semantic implication in all situations. Such work is mainly conducted only with the staff of pen and paper. The latter includes an account of computer application utilized in terms of selection of software, and network. Thesaurus, like other lexicons, is an open ended wordbook consisting of hundreds of thousands lexis alongwith other required information, viz. grammatical information. The compiler has to select maximum possible lexical items and their semantic shades having multidynamic dimension such as dennotative. connotative, figurative, referential, and objective. Here, in this research work, the selection has been done with restricted data containing the lexemes belonging to the different subject-domains and senses, and sub-senses used commonly and frequently among Hindi speakers. The subject-classes that are not taken into contemplation in selecting words here include neologisms, proper name, archaic words, pure Sanskritised words, dialectal words, and lone or borrowed words.

Gradation with higher step of process of compilation sorts out the material unworthy of compilation. Hence, the vocation of grading of the material being used for compilation is processed to finalise it. The selected material (lexemes and their meaning-shades) is difficult to compile in the lexicon. Therefore, one has to possibly make a choice about those lexemes and thier semantic shades being included. The graded material is represented in the mode of pre-well-defmed assorts. The present research precisely intakes the representation comprising the feeding of data (i.e. adequately selected and graded lexical items alongwith their meanings and grammatical information) for promised output. The sense representation can be classified either in the individual mode or globally. In the case of individual mode, the senses of particular lexical entry and in the globle mode, the subject-domains are taken into account.

Preparation of thesaurus has to settle some practical problems of compilation such as variation in spelling, ordering of grammatical category, numberization of semantic assignments. nature of cross-reference, and indexing system. The peculiar feature of language is varying in spelling but having the same meaning(s). Such variation is found due the variety in pronunciation of words of a particular language spoken by a segment of the speech community. Like other languages, Hindi has also some variations. For example, y and j are alternants as in the Hindi

words vogya which can also be written in its alternative as jogva and likewise, as in vamunaa and its alternative lamunaa. Such varitions are considered here as free entry by showing equality in semantic assignment. But for machine-readable compilation, it is tackled only by feeding variants and by marking equal (x = y means x is equal to y) in programme. Here, x stands for one lexical item and y for another.

Like other devices, numberization is essential one because it helps the readers to understand the meanings classified in more than one sense-group according to their approximate assignment. The manual thesauri or other types of lexicons require it only to show or distinguish between approximated meanings, whereas for electronic thesaurus it is not essential except only in terms of codification and marking record numbers. The cross-reference is used to nest the whole body of dictionary/thesaurus. It has two types : proceeding (•*—) and preceding (—•). The first one help the user to find the reference of that entry which occurs on later page(s) or on the same page but should come afterwards. The second, unlike the first, helps the user in searching backward reference from the original point. The cross-referent can be given at any level of entry according to necessity. As markers, the following devises and signs are used : see. c.f, <number>, and some other kinds of hints. The cross-reference marker is always given within the small brackets.

The index device is used to find out the location of respective entry being searched. For manual purpose, it can be represented into the following manner.

1. Giving all pp (page numbers), e.g. X (name of lexical unit) 1.5.10. . . . .

2. Marking only the first occuring pp X x (i. e. any number of pages where the reference of lexical unit is being searched) and then going through cross-reference device to the respective place. It helps the user to enter into the body of text. But for electronic use it is not taken into formation except in case of indexing the entries which are done automatically during data feeding.

The suitable approach aids to direct the research work in proper manner with appropriate output. Here, two types of approaches are followed in compilation. They are associative semantic approach and free-form-entry approach. The semantic thesaurus is intricately related with classificatory system where there is a contextual consensus conveyed through a lexical unit dominated in meaning arrangement. Obviously, each lexical item has its own semantic flavour and sphere, but (it) also shares some countenances with others having their own value. According to the degree of approximation in sharing features they are put into a group or into a system.

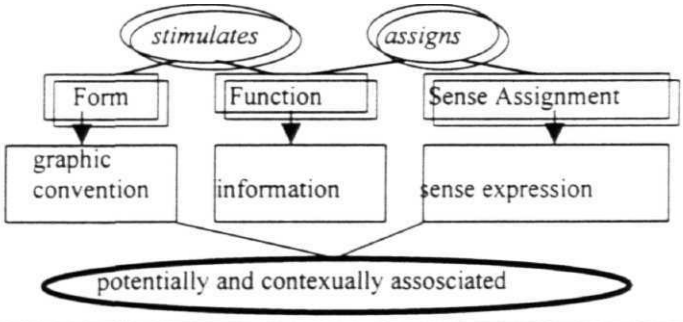
The lexical items being included in the thesaurus should be arranged as free lexical entry rather than by making sub-entry which enables the manual or electronic compiler to utilise and to process congruently. Katre (1965:21) made a satement where he focused on lexicon in general but it is also equally applicable to thesaurus. He rightly says. "The lexicon is an inventor} of the free forms of language arranged systematically, and against each form are shown their functional load of meaning in each distinct meaningful situation."

2.5.2. **THESAURUS** CONSIGNMENT

As one can notice from Katre's (1965:21) statement, the central node of the thesaurus is the semantic representation to be taken into account to consider the meaning representation. The semantic expression as an essential component of lexicographic work comprises the multi-dimentional usage of meaning of a particular lexical item. The semantic attributes (meanings) of a lexical item are precisely framed according to the order of nearest to nearer in order to show the

frequency and precise usage of that particular lexical item in the specific context. Though most of the thesauri as well as dictionaries have been compiled in order to provide an equivalent or synonyms of an entry rather than being intended as a way to outline the concrete picture of sense-assignment. The equivalent/synonyms are given according the possible contextual usages of that lexeme. But the compiler can never be able to give the same sense-effect through equivalents/synonyms because "The semantic range of a lexical item, as widely accepted, is inherently embodied in such a way that the replacement of it by some other lexical items, whether it is synonyms or equivalents, cause the space between the original usage and the substituted one. Transfiguration of only certain semantic features of the lexical item which is filtered out in the substituted lexical item having its own sphere comprises with particular semantic values which eliciting the inconsistency between them atleast in one sense-representation" (Pandey, 1993). The thesaurus is a unilingual work where the task of compiler is to overcome such problem by taking an account of Katre (1965:36) suggestion, viz. "If the lexicographer is operating with a unilingual lexicon he has the additional responsibility of supplying accurate definitions ... The definition should really cover the content of the word and not contain either more or less than the content; such a definition must be able to take the place of the word in the chain of utterance without any change in meaning."

The whole substances of thesaurus entry consisting of all exhaustive relevant information according to its application are related coherently in unique pattern, which can be divided into three broad categories - form, function, and semantico-pragmatic implication. The form is related to the physical property (i.e, graphical convention) which stimulates certain functions dealing with the information of the characteristics of the lexical item, viz. the detail about pronunciation, grammatical and usages, assigning the concrete sense refers to the real sense or idea which enables the reader to get the real picture intended to convey through that particular lexical item in the given context. The whole argument can be viewed in terms of the following diagram:

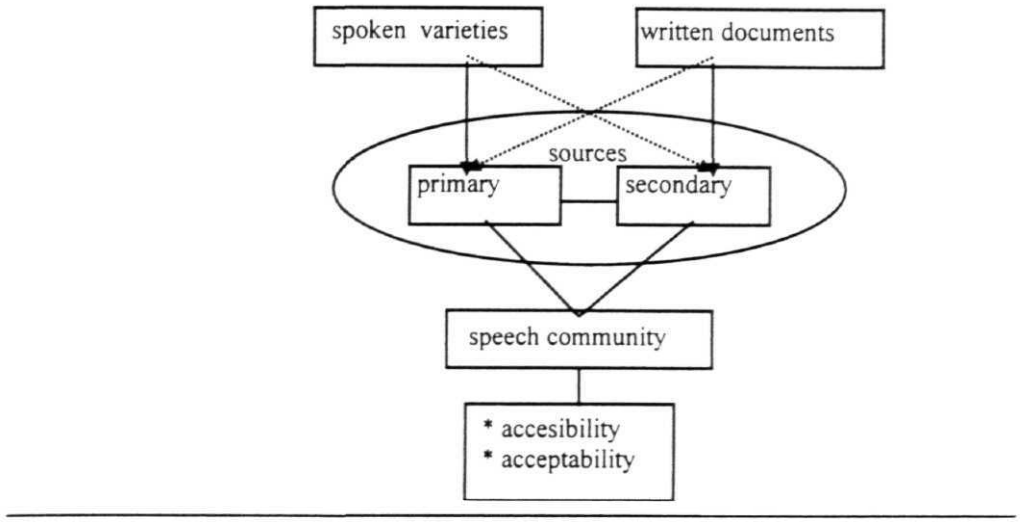


Digram 9 : Represention of the Unique Relationships among the Form, Function, and Sense -- Assignment of a lexical entry.

Likewise, the relationship within the lexical item can be drawn in terms of labelling just as the linguists and lexicographers have designated to the different part of the lexical item. The following diagram represents the inter-unique-relationship within the word. Here, we can see that the Lemma functions at the head of complete lexical family which governs the grammatical category because of its own nature whereas the sense representation (which are mostly either synonyms or antonyms) is governed by the grammatical category by controlling and assigning the same category to which the Lemma belongs. In fact, as far as semantic assignment is concerned, the semantic representation vary due to the different usages of lemma.

A critical evaluation regarding the data selection and the fixation of the sources is a prerequisite in planning a thesaurus. Though the thesaurus itself is a vast compilation, there are many parameters and criteria set in order to evaluate the nature and the range of the subject being included

in the thesaurus. There is a *bistrategic methodology* which is followed in evaluation and selection of materials. They are primary and secondary sources and their interrelationship to the speech community. Therefore it is triangular relationship in the bistrategic formulation. They can be represented in the following diagram :



Digram 10 : Data Sources

The primary source, as widely used, is the spoken varieties. The secondary sources is written documents available. Indeed, the seconadary source shows a resemblance with the primary source. The primary and secondary sources are evaluated by the speech community in terms of the parameters of accessibility and acceptability. Sometimes, the primary and secondary sources depend upon the nature of the dictionary. For instance, in the case of historical dictionary, the written documents and manuscript will be the primary sources, whereas in the current dictionary, the spoken variety will be the primary source.

The synonymous thesaurus is based on the idea of complete and partial synonyms which are assessed by their usages. Now. the lexicographers begin with a belief that there are such complete synonyms which can be replaced in all environments. The idea of complete synonyms goes against the conception that two lexical items can never be *same* in terms of orthography and semantico-pragmatic assignment but can be *similar* by sharing maximum contextual features. Recall the statement of Ghatage (1973:27), who notes that "Linguists speak of the impossibility of perfect synonyms in a language and base this view on the fact that the total distribution of any given vocable is bound to differ from that of any other in the language and hence there cannot be a replacement of one by the other in all the environments. Without such a possibility we cannot speak of perfect or complete synonyms. Others will be satisfied with a replacement in the majority of the environments or in the most significant ones to be able to draw the conclusion that the two items are synonymous. In this case we may speak of near or approximate synonyms". The representing range of synonyms is also a parametric question among the lexicographer. There is also thus other question as to whether the synonyms may be a single individual lexical unit or it may be a compound. Sometimes, they even think over that phrases can be synonyms or not. Though it is apparently a matter of orthographic representation, but there is semantic uniqueness underlying in it. For instance, if there is a phrase consisting of more than two words assigning the same meaning which is assigned by the an individual lexical **unit, it will be** considered a synonym. In fact, the term synonym itself is a sense based term but **not** orthographic representation.

The systematic representation of the synonyms and the semantic classes provide a clue to make the distinction between the two synonyms at the macro level as well as micro level. Hence. By semantic reference, the synonym itself becomes a part of the hierarchical semantic representation. In the case of the lexical innovation or coinage, some processes are adopted. Such derivatives caii be identified as complete, partial and opposite in terms of original semantic preservation . In the case of complete semantic preservation, The lexical item derived from other elements retains the same senses assigned by the original elements or vocables. In contrast, in the case of partial, the sense will be partially sustained, whereas in the opposite case, the original sense will be completely lost. This can be traced out as lexical dependence tracing the origin of different forms. Hence, the source elements sustain their individual entity. Each derived vocable is an output of the semantic development as a distinguished lexical entity. They can be considered as independent elements if no attempt to relate the different senses is successful. If however we can show how one sense is related to the other as a result of semantic development we will consider the items to be basically the same and regard it a case of polysemy.

2.6. **THESAURUS COMPILATION**

The compilation of thesaurus, like other models of dictionary and lexicons, has various modules depending upon the purpose intended. Thus there is no static formalism in thesaurus compilation. But the nature and representation of items show sense identity in their aids and aims. The philosophy of building thesaurus depends broadly upon the proposition of the language specific representation which is governed by time and space and requirement. Here, two inclinations of thesaurus building viewed in terms of oriental and western compilation are highlighted. These two trends have their own distinguished and idiosyncratic tenors in building the thesaurus which are given below.

2.6.1. ORIENTAL COMPILATIONS : ***A REVIEW***

The lexicographic tradition in India is very glorious in its historical perspectives and one can trace this development from the Vedic literary era to the modern period. Formation of the compage-network and subject which are related to the *lexical dictionary* dealing with "lexis" in the synonyms and polysemous forms and *non-lexical dictionary* pertaining to the conceptual information about the respective subject represented in an encyclopaedic account are multipartite in its infrastructure. The earliest lexicographic work based on the different principles and criteria was compiled and embellished by making special notes on individual lexical items in the vocable-grouping-format consisting of expressions which were taken into account in order to manipulate the contextual usages adequately. For instance, the 'nighanTus' (Sarup, 1967), as afflatus-compilation and provative principiant lexicographic reference, were affined in Vedic tradition. It consists of the lexical items belonging to some semantico-pragmatic sense-representation. For example, see the following stanza consisting of twenty one names for earth":

Om gau| gmA| jmA| kshmA| kshA| kshamA| kshoNI| kshitiH] avaniH| urvl|
prithvl| mahl| ripaH| aditiH| iLA| nirtritiH| bhUh| bhUmiH| pUshA| gAtuH|
gotrA| ityekarnviSatih prithivInAmadheyAni ||1||

Later this pattern was adopted as the basis and source to the Yaska's niruktA. In the 4th Century B.C. of compilation-span, another distinguished compilation came into being namely the Amarkosha - a semantic/thematic thesaurus, written by Pandit Amar Singh, having encompaged in verse-form with a view to memorise it. For instance, see the following stanza which consists often synonyms

for 'death'. They are *pancatA, kAladharma, dishTAnta, pralaya, atyaya, anta, nAsha, mrityu, maraNa,* and *nidhana.*

syAtpancatA kAladharmo dishTAntah pralayootyayaH |
anto nASo dvayormrityurmaraNam nidhanoastriyAm ||

Hence, entries are arranged according to multifarious semantic domains and multilateral subject representation in multipartite network. Evry lexicon has its own format, which may be or approximately identical to the others, posed out according to its prerequisites - application and utilization. According to the Vedic literary study, the Indian tradition of lexicon came into existence as the Vedic lexicon - 'nighanTu' framed out on the basis of "niruktA". Likewise. Hindi lexicons makes their stand-points, showing unlikeness against the strong metaphoric claim of Bahri (1980:41). regarding the position of Hindi lexicographic works, who points out. "The story tells a tale but lends no guidence". But, there are certain hints of guidence underlying in them which can be comprehended as per aims and objectives of the lexicon being referred.

The latest thesaurus of Hindi is Arbinda Kumar and Kusum Kumar's "samAntara kosha" published in 1996. Although the present electronic thesaurus and Kumars work were going on for several years in parallel, there had been no input from one to other because of lack of knowledge of the other effort. "samAntara kosh' is divided into two volumes : "sandarbha khanDa'- this is the main part of the thesaurus consisting of the lexical entries antd their associated information. The second volume is 'anukramak khanDa' - i.e. the index of the entry. The index volume is bigger than the text volume. No doubt, this is a major lexicon in the span of the modern dictionary making, and in the context of Indian languages, the '"amarakosha". this can be marked as a milestone in modern thesaurus making, in terms of design and development. Some important features of the thesaurs can be highlighted here:

- This is a thematically classified thesaurus covering 1,100 head-titles along with 23.759 sub-head-titles. As far as the expression is concerned, it consists of 1.60.850 expressions.

- The classification of classes is based on the flow mode, they have followed the grouping system. But some times they do not. For example, "parvata ghATI" is in the 16th class whereas "pASANa" is the 99th class and in between he has used the "kAla" which shows the discontinuity in serial representation.

- It can be referred to either from the class-contents or from the index.

- As far as the content is concerned it is very rich in providing the synonyms, related words as well as classes.

This is used as the source of the current research with completely different classifications.

2.6.2. WESTERN COMPILATIONS : *A REVIEW*

The history of western compilation has had its own outstanding products in the thesaurus building. Here some are selected only to assess the modules in terms of their merits and demerits. With the fast technology, the western compilation has obviously had more advanced designs rather than that of the oriental. The designing concepts in the western trends are representatives for the modern model. But still it requires some reformed downright configuration system. It should be noted that the detail given in the following citations might be editor's claims as mentioned

somewhere in the thesurus which is followed by the critical comments. They are both alphabetically classified and semantically classified thesauri. Some models are cited below just for example :

THE OXFORD THESAURUS

The features and compilation strategy of **"The Oxford Thesaurus** : **An A-Z Dictionary** of **Synonyms"** are highlighted in the following paragraphs.

Features :

- Easy to use and comprehensive.
- 215,000 synonyms.
- Example sentences for every entry.
- Fully indexed.

The Network of Compilation

- It was planned in 1986 as a 300,000 words volume but was consequently increased upto 650,000 words (approx.) without index. It embodies quite comprehensive information on . examples of an entry as well as ideomatic expressions and Index of words and expressions.

  - It has quite unified networking features, viz attributes, selection, and representation. The compiler has followed the following criteria in selecting headwords. They are :

    - Frequency
    - Low frequent words as per requirement.
    - Common derivational listings are not given, in order to avoid duplications. Sometimes, nonautomatic and semantically obvious derivations are not taken into account.
    - Sometimes, words having identicality. are set-up as separate entries due to the difference in detail and application.

  - It is facilitated with the index, where the words are listed in alphabetical order. Moreover, the phrases are also listed in their respective entry.

  - The entries are arranged in the alphabetical order, and therefore, searching can be done either from the text or from the index.

  - It shows the frequency of occurences and contexts. The phrasal verb and phrasal idoms are arranged under the same strategy. All words and phrases are arranged in letter-by-letter alphabetical order.Plural form of Headword are also used as sub-headings. A sub-heading is listed in its respective postion in the main entry.

  - The illustrations are given in **terms** of the form of contextually appropriate sentences used as examples.

  - As far as the synonyms arrangement is concerned, the compiler followed the following criteria:

* Nature of Arrangement: Syntactically, sematically, and idiomatically.
* The semicolumn (;) is used to make subgrouping of synonyms.
* Transitive vs. intransitive / literal vs figure of speech usage, etc.
* Definition also used to explain a synonym.

- In the case of Cross references, the following points are considered:

  * Index is used as a cross reference mechanism.
  * Very few cross references are given in the main text.
  * Modes : numbering, see also
  * Direction indicator: below and above.
  * Apporach : synonyms to main entry.
  * (o) itself functions as the cross reference in multi-mode application.

- It provides some labels which make the understanding of the usage-context of the particular word easy. They are used with the following labels showing the nature of headword.

  - Usage labels : colloq (for colloquial), slang, taboo, archaic, old-fashioned, technical. literary, etc

  - Regional labels : Brit (for British), US (for United State). Australian. Canadian. New Zealand . etc
  - All labels occur in combination.
  - Usage labels always take precedence over regional ones.
  - The main entry and idioms are not labelled.
  - Two regional labels do not apply to the same entry.
  - Regional defferences are labelled for headword.
  - No regularity in labelling synonyms.

- The Spelling Variation and Substitutability are also taken in account alongwith some other points given below:

  - Nature : contextual substitution
  - Idiomatic and syntactic adjustment.
  - Two sets divided by (;) are given by contextual examples.
  - The familar synonym does not required example.
  - Propositional and adverbial particles are highlighted by bold, and so on and so forth.

COMMENTS

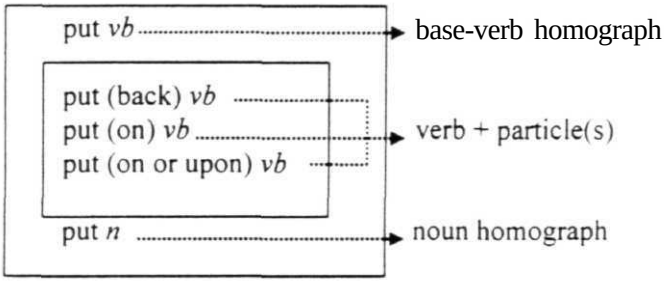
The Oxford Thesaurus is. of course, an exhausive and well-defined alphabetic thesaurus. It is arranged as a dictionary of synonyms. But still in one way or the other, the Editor has followed the semantic determination of entries which are overtly marked in the text. The whole book is devided into two parts - the first part is the text and the second part is the Index. From the functional and applicational points of view, the Index has major utility. The volume of the Index is as large as the text. The fundamental requirements of the manual thesaurus are available in configurational style. The representation of the entries is principle-oriented. The search facilities are available at both points - text and index - to enable user to find out the contextual application of the head words.

The features mentioned above are not sufficient, so far as concerned of the design and the nature of thesaurus. It is quite true that there are two types of thesaurus, as mentioned above, but due to the applicability of thesaurus, the user seeks the following facilities : The overtly marked semantic classes which define the domains and context and the minimum effort in the searching mode. But based in these issues alone, a thesaurus does not achieve the merit of accomplishment.

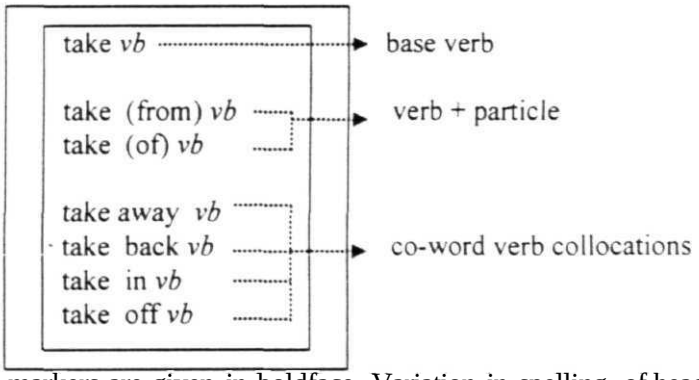
THE WEBSTER'S THESAURUS

As far as thesaurus is concerned the **"Webster's Collegiate Thesarus"** is also widely referred reference book. The main source of this thesaurus is the Webster's Third New International Dictionary published by the Merriam-Webster (R), USA. Its main features, extracted from the Preface, are highlighted in the following paragraphs:

The Webster's thesaurus is arranged in the alphabetic order. The order of entries is designed according to main entry and sub-entry scheme. The homograph headwords has been entered in historical order. And in addition to this, the base-homograph is followed by its particle combination.



Fixed verb and adverb collocations commonly entered in the dictionary as TWO-WORD verbs having boldface at their appropriate alphabetic order. Example :



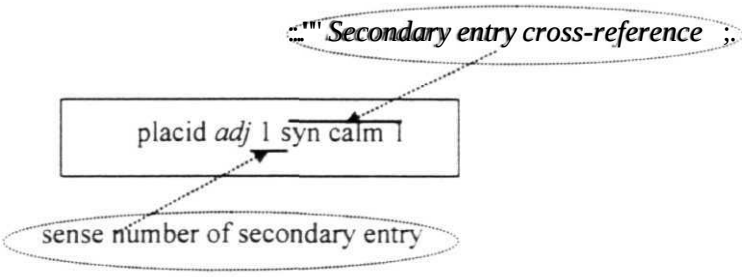
Singular and plural markers are given in boldface. Variation in spelling of headword is given in the small bracket. The following usage classes are considered as main usage.

syn > synonym
re I > related word
idom > idomatic usage
con > contrastive word
ant > antonym,
etc.

Numbering procedure followed in distinguished the sense-group. 'Meaning core" is supplemented by a usage note introduced by a light face dash when addition, information or comments on syntax or usage are required. Sometimes, the interjection is directly given with meaning core. Each meaning core is followed by a verbal illustration enclosed by angled brackets. Sometimes, double illustration also selected. After the meaning core and verbal illustration the 'syn' is given in the line below. There are two ways to present the cross-references:

1. 'compare'
2. compare (any number specifying the referring sense, eg 1,2,3,4,5,...)

Secondary Entry formation followed in drafting this thesaurus is given below :



THE ROGET'S THESAURUS

The world famous semantically classified thesaurus **"Roget's Thesarus of Synonyms & Antonyms"** has been used in the world over as the only available thesaurus for a long time. The Roget's thesaurus have been published from P.S.I. Associates Inc. USA in 1988 comprising of 1000 entries (treated as semantic classes) with 100,000 words. Its features are sematically classified. The semantic classes and sub-classes are as follows :

Classes

I. ABSTRACT RELATION

Section

I. Existence
II. Relation
III. Quantity
IV. Order
V. Number
VI. Time
VII. Change
VIII. Causation

II. SPACE

Section

I. Generally
II. Dimensions
III. Form
IV. Motion

III. MATTER

Section

I. Generally
II. Inorganic
III. Organic

IV. INTELLECT

Division

I. Formation of Ideas
II. Communication of Ideas

V. VOLITION

Division

I. Individual
II. Intersocial

VI. AFFECTION

Section

I. Generally
II. Personal
III. Sympathetic
IV. Moral
V. Religious

The model of the present work is to some extent is related to this. However, the present work provides much different scheme of classification and greater amount of semantic information. It does not merely give the synonyms or antonyms.

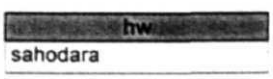
2.7 ATTRIBUTE TO ENTRY FORMATION

The entry formation is a task of representation of the lexical item in the appropriate format in the lexicon. In Dictionary- making, this task comes at the second level which covers the following task: How the informations of a lexical words be allocated in a defined format. Here the following criteria have been followed to decide the attributes and the coverage for the present work:

1. Selected purely Hindi words.
2. Word having more than one meaning.
3. Words were grouped according to their context and semantic assignments.

Here a brief account *of* each attribute followed in the present work is given below with suitable example selected from the database prepared for the present work.

HEADWORD : The headword is a word which is being searched. The other attributes of the word are associated with it. For instance, all 1291 headwords shown in the appendix 3 are headwords.



GRAMMATICAL CLASS : It represents the syntactic information of the word. Here, only four parts of speech, viz. noun, verb, adjective and adverb are taken into account, the grammatical class of the word sahodara 'brother' would thus be as follows:

gc
saM

SEMANTIC DOMAIN : The semantic domain covers the world knowledge into eight domains. Normally it covers the information about the self and external physical or non-physical world. Moreover, other information related to religious, measurement, affection etc. are also taken into account within these domains. All these eight domains have been classified into subclasses in the Chapter 3. The semantic domain of the headword sahodara "brother" is 's' (= SELF):

sd
s

SEMANTIC CLASSES : Semantic classes are nothing but the sub-classes of the semantic domains. It is noted here that the semantic class refers to contextual value of the word and it also assigns its thematic value. The semantic classes have been identified, in terms of processes, consequences and objectives, both the physical and non-physical entities of the world knowledge. Obviously the classification of semantic classes followed here is based on the universal outlook and language specification. The classification of semantic classes have been given chapter 3. The semantic class of the headword sahodara 'brother' is :

sc
self.existence,theoretical,conditions,contiguous,brotherhocxd,brothers,own

SYNONYMS : When a word is used in place of another word of the same langauage. it is said that word is synonym. This conception leads the following points:

1. Synonyms are referred to in the monolingual context.
2. It shares the semantic or thematic values assigned by the concerned word and its substitution.
3. It is also noted that each lexical item has its own entity in the sense of usage. Therefore. synonyms can be partial synonyms or complete synonyms. In the case of partial synonyms it covers not all but few features of the word whereas in the case of complete synonyms there is complete sharing of almost all features *of* word. Thus they are also distinct and nearest synonyms.

sn1	sn2	sn3	sn4	sn5
bhAI	agraja	anuja	bhrAtA	bhaiyA

ANTONYMS : The words having opposite meaning in its physical characteristics, quality, and orientation. Antonyms carry both positive and negative qualities of the word in question. Antonyms by nature indicates two words in contrasts. For example, antonyms of sahodara "brother" would be :

ant1	ant2
sahodarA	sautelA

COMPOUNDING : Association of two words irrespective of its grammatical catageroy constitue a composite meaning and this process is called compounding. The words in association are called

compound words. Such words in association can belong to the same grammatical catageroy or different classes. The meaning of a compound word reflects any one of the two or more words in association or an absolutely different meaning apart from the word meaning.

The word in association can have following combination of grammatical categories : noun-noun, noun-verb, noun-adjective, verb-verb and etc. The compound words like noun-noun can have a meaning which is not local meaning of any of the word, which has some third meaning, e.g. nll-kanTha literally means a person having 'blue' coloured 'neck'. But in this context it means "God Shiva" which is a third meaning. The compound words having verb-verb combination gives a composite meaning out of the two words in association. For example, AnA-jAnA which means 'to come and to go' in general. But in this context it has the meaning of communcation also (i.e. yAtAvAta). Let us now consider the compond word combinations with 'bhAI'.

com1	com2	com3	com4	com5
bhAIbaMdhu	bhAIbhatijA	bhAIbirAdarI	bhAIdUja	bhAIcArA

ROOT: Root is the basic form of the word which cannot be divided any further into some meaningful entities. In the present work the root has been taken account in the following ways :

- The word which cannot be further divided into a meaningful entities in Hindi, is of the first type.e.g. bhaai is one such example.

hw	root
bhAI	bhAI

- The word which has the dictionary entry as minimum meaningful unit. i.e.. the resultant word after deducing the prefixes or sufixes. e.g. laDkA is a root but not laDaka. even though.the construct laDakapana 'childlikeness' the component laDaka can qualify to be a root. For example,

hw	root
buddhimAnI	bhuddhi
bhaiyArI	bhAI

DEFINITION : Defining 'definition' is the precise task of expression covering the complete characteristics of word or terms in a precise manner. Although physically it is very short but in expression it is very compehensive.

defin
apane hI mAtA-pitA ke putra ko bhAI kahate hai~.

ILLUSTRATION: Illustration is one kind of expression which resolves the complexity in simple and explanatory fashions. In this thesis illustration covers the following points:

1. It gives an explanation of the syntactic feature,like - gender, number, person.
2. It provides the etymological explanation, and
3. It gives the contextual clarifications.

illus1	illus2
bhAI choTe yaa baRe ho sakate hai~.	hama-umara laRake ko bhI `bhAI' kahate hai~.

EXAMPLE : To make the task of illustration a simpler example is the best option. Here, the object which is defined and illustrated is designed physically . Thus, the object is displayed to support the illustration.

examp1
surendra aur mai ek hI mAtA-pitA ke beTe hai~ isa liye vaha merA bhAI hai.
examp2
shailendra ham- umar hone ke nAte vah merA bhAI hai.

DERIVATION : Derivation is a word derived from the stem or base by adding the derivational and inflectional affixes to it.

hw	drv
jyAna	jyAnapurvaka

COLLOCATION: When a word mostly occurs with a particular word, thus creating a set combination, it is called collocation. But this togetherness is constrained by some grammatical rules, and sometimes it violates certain existing rules. Thus, collocation is something in between the grammar and the pragmatics. This statement is further proved by another author's argument elsewhere that "'Hence collocation should be considered a phenomena between grammar and context" (cf. Singh 1980:134). For example, cilcilAtl dhUpa "scorching sun". There can be three types of collocations:

1. meaning oriented
2. semantic range oriented
3. neither of the above

hw	coll
badacalana	aurata

RELATED WORD : A word which is associated with another word, they are said to be related word. For example, the related words of sahodara "brother" are :

rw1	rw2	rw3	rw4
parivAra	sambandha	bhAIcArA	sagA

SPELLING VARIATION : A word is represented into two different forms with some orthographic variation. For example.

sv1	hw
mA	mA~

IDENTICAL WORD: When two words are identical in their sense assignment with minimum variation in the orthographic representation, they can be said to be identical words. For example. barkhA and varSA both meaning 'rain'. This would be in contrast with near homophonous words that mean different things, such as divA 'given' and dlvA "lamp". Like the spelling variation, the identical word is also equal to the headword and its spelling variation because it shares the all sense-references. How they are interlinked for the retrieval of information and for the identical word would be shown later.

2.8. SUMMARIZING NOTES

In summary, one may recall the oft-quoted statement of McNaught (1988:26) on the objectives of computational intrusions into language analysis vis-a-vis lexicon-building : "CLing (Computational Linguistics) should be paying more attention to the lexicon, and especially all the more so as there are strong movements within CLing towards the adoption of lexicalist theories of language, which imply rich dictionary information, and indeed the dynamic use thereof, a notion which is in stark contrast to the traditional view of the lexicon as a static, monolithic reservoir of lexical data, having no interesting structure, being merely a necessary appendage to the grammar."

CHAPTER 3

SEMANTIC DOMAIN-ASSORTMENT

"Families of words are like families of people .... that the family has grown a long time, that each term has gone its own way, and no one of them is exactly like any other. A word may be able to take over the job of a word similar in sound or meaning, but it may not. "

- **Laird** (1990:ix).

3.0. PREPARATORY REMARKS

"A word may be able ... but it may not." - the uncertainity expressed by Laird vividly depicts the problematic of a thesaurus-builder. The basis of any thesaurus is that it should enable its users to look for suitable replacements. It turns out the equivalents which are not only difficult to hunt for but that they are often almost impossible to arrive at. The entire edifice of a thesaurus breaks down, if these are not possible to achieve. However, intuitively, we all know that such substitutions and replacements we keep on aiming and arriving at in our daily paraphrasing work more often than not. Therefore, no matter how much uncertain (or even inglorious) the search may be. it is still worth giving a chance, and hence thesaurus is worth-building.

If thesaurus is a kind of lexicon embodying the maximum possible semantic information on each lexical entry in the form of sense-assignment represented through synonyms and/or antonyms as well as definitions or even related and associated words, the design framed in compiling thesaurus is bound to vary in nature depending upon one's semantic outlook or on the purpose for which one is readying it. To test the ideal criteria of semantic countenance in the thesaurus, they are sorted out by application of semantic principles as given in linguistic studies. The central node of a thesaurus is the semantic representaion of lexical items taking into account the vouchsafing of its semantico-pragmatic references. Lexical meaning plays the central role in lexicon compilation. The lexical meaning comes into attention with a view to understanding the situation - the surrounding as well as the involvement of the speakers and hearers where the expressive reference is being transmitted through the linguistic codes. Obviously, there are some contexts, viz. external situation, mode of speaker's intention and hearer's perception which lie in the same socio-psychological status. By assembling these facts, Zgusta (1971:21) had classified the context as 'verbal* - reading or hearing of the expression and 'situational' - a person's activity in that particular situation, space, and time. Though he himself accepted them as clearly segregated classes there is some degree of overlapping between them. But, they are factually, rather coherently interrelated; where the previous one functions as a medium and the latter one as an enforcing agent. The situation makes one understand the actual meaning of a word carrying explicit information.

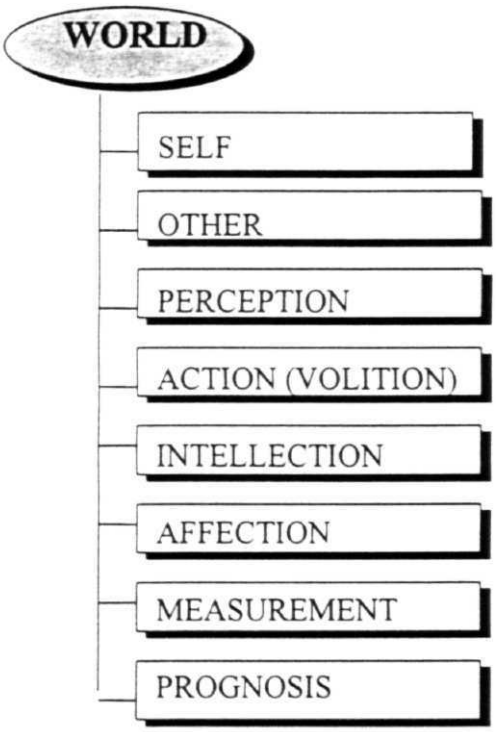
3.1. WORLD KNOWLEDGE : *SEMANTIC APPLICATION*

Every lexicon has its own model semantic classes and its own method of entry formation, especially the thesaurus having two models -- alphabetic and semantic (as discussed previously), depending upon the nature of arrangement of lexical entries. It is. obviously, the physical part of the lexicon dealing with the visible drafting shape resulting as an output and the non-physical part which undertakes the whole process taken into practice as an input issue. The whole vocation of preperation of a semantic thesaurus undergoing a series of tedious contemplative activity ultimately resulting in assemblage of attributes, undersigned processes, drafting devices, and exemplary approaches which are viewed in elucidation.

A semantically classified thesaurus comprises some hierarchical categories which are based on both intuitions and assumptions about the world at large. Now if the Sapirian dictemit is true that different languages or groups of languages live in distinct worlds, not strictly comparable, it will follow that there must be different (but, could be related) designs of thesaurus valid for a given group of languages or their cognates - both geneological and structural. The assumption being made here is that the present design should hold good for South Asian languages - particularly those belonging to the Indo-Aryan and Dravidian families - which have shown a great degree of linguistic convergence at various levels.

3.2. SEMANTIC DOMAINS

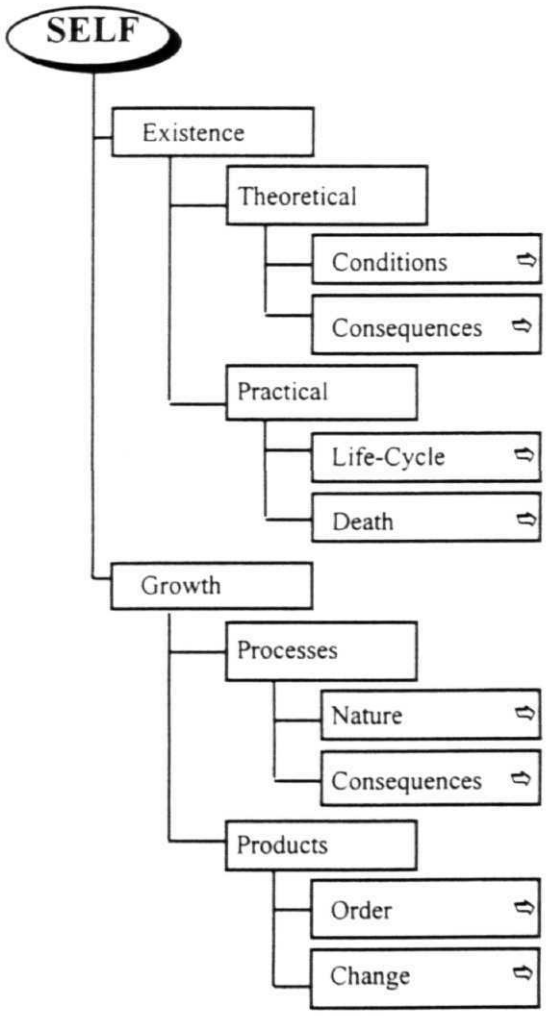
The pivotal attributes of entry-formation can be seen below: The lexical entries, being included in a semantically classified thesaurus, share some general features which are identified subject-wise by entitling their respective subjects. Therefore, it consists of the major sense of the word or words to which it belongs. For instance, these could be a word or a word-sense which belongs to literature, technology, science, or commerce. The broad idea behind the semantic domain as presented here was outlined by Singh and Pandey (1996) in terms of eight sense-groups as given below:



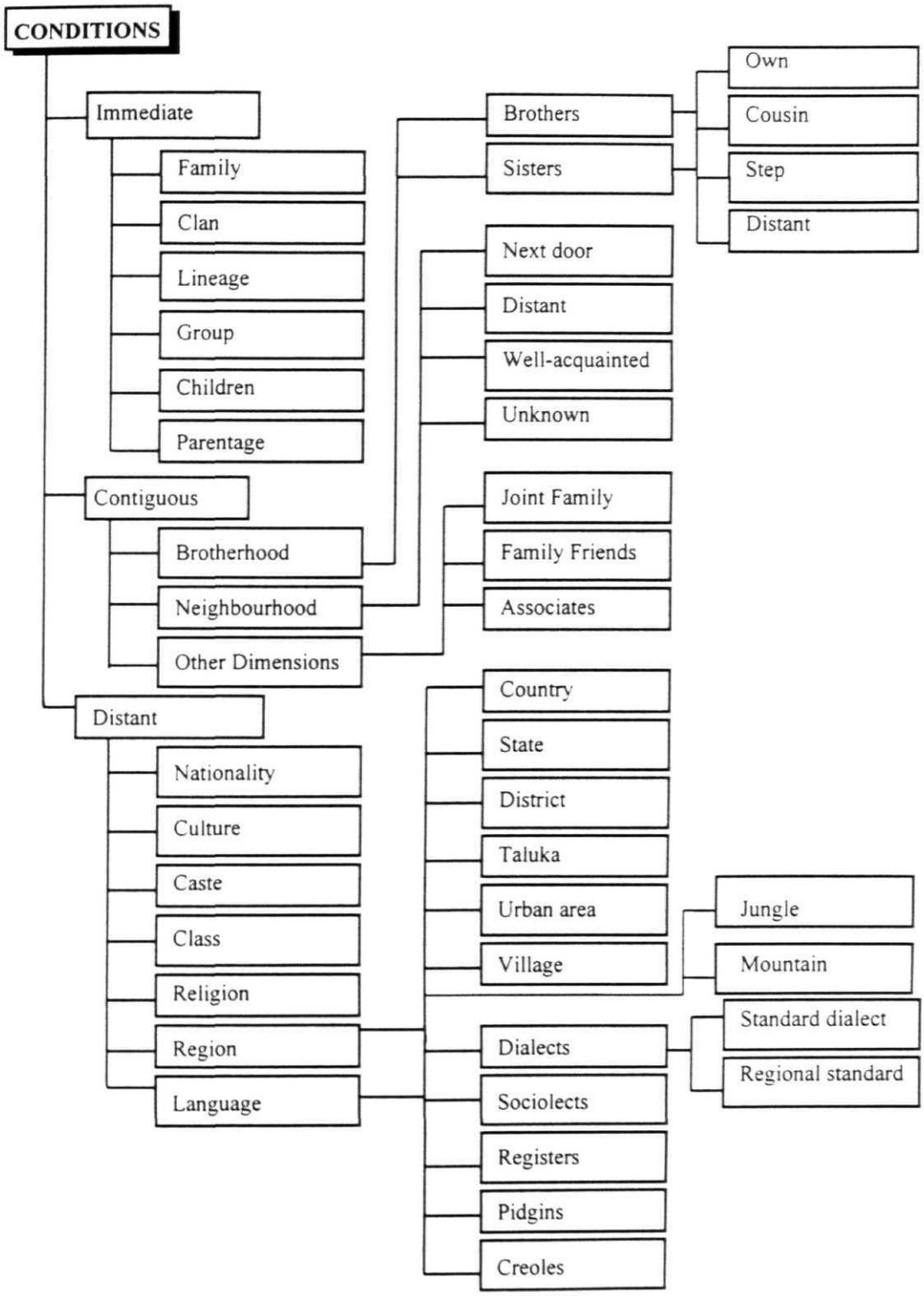
The semantic world, as divided into eight classes/domains, is comprehensively represented in determining the context and component of the word from where it belongs. Indeed, the classification is logical and philosophical in its constitution. The scientific approach or rational approach shows up in the following classification. As stated earlier, it is a language-specific classification. But, moreover, the universal features are also found in this spacious design. There are certain parameters which justify this classification of the semantic world into eight classes. They are discussed in the following paragraphs. It must be borne in mind that the proof of its utility can be gauged ony by running the thesaurus programme which has been prepared based on this hierarchical classification and which is enclosed in the accompaning floppies.

It is quite possible that two words will share common features and it is also possible for a single or a group of lexmes to have multi-entries or complicated cross-references. The grouping system can be done depending upon the correlation of entries and semantic features. This uniqueness correlation of feature in each case helps the lexicographer involved in the work of compilation to detect the contextual appropriateness of a word's application. The issue of semantic function as well as pragmatic characteristics can be gauged through the correlation between two objects or things. Assortment of semantic domains here is based on the criteria of relationship, association, condition, consequences, and processes from the universal as well as language-specific points of view. These eight domains could now be expanded in detail to show how exactly the world has been conceptually structured (or atleast how it is assumed to be structured) according to this thesaurus. Each sub-class or its further division has been suitably formed in order to avoid any confusion. These classes under each domain can be seen here as well as on floppy disks accompanying as an appendix to this thesis (using Windows'95 platform and Visual Basic 5.0). Of course, it would be ideal to give them in terms of a tree diagram showing the total picture of this classification at a glance, but that is not possible because of obvious limitation of space. Instead, to start with, one could look into the broad divisions as given below. Here the arrow (^) shows that the class has been furher classified.

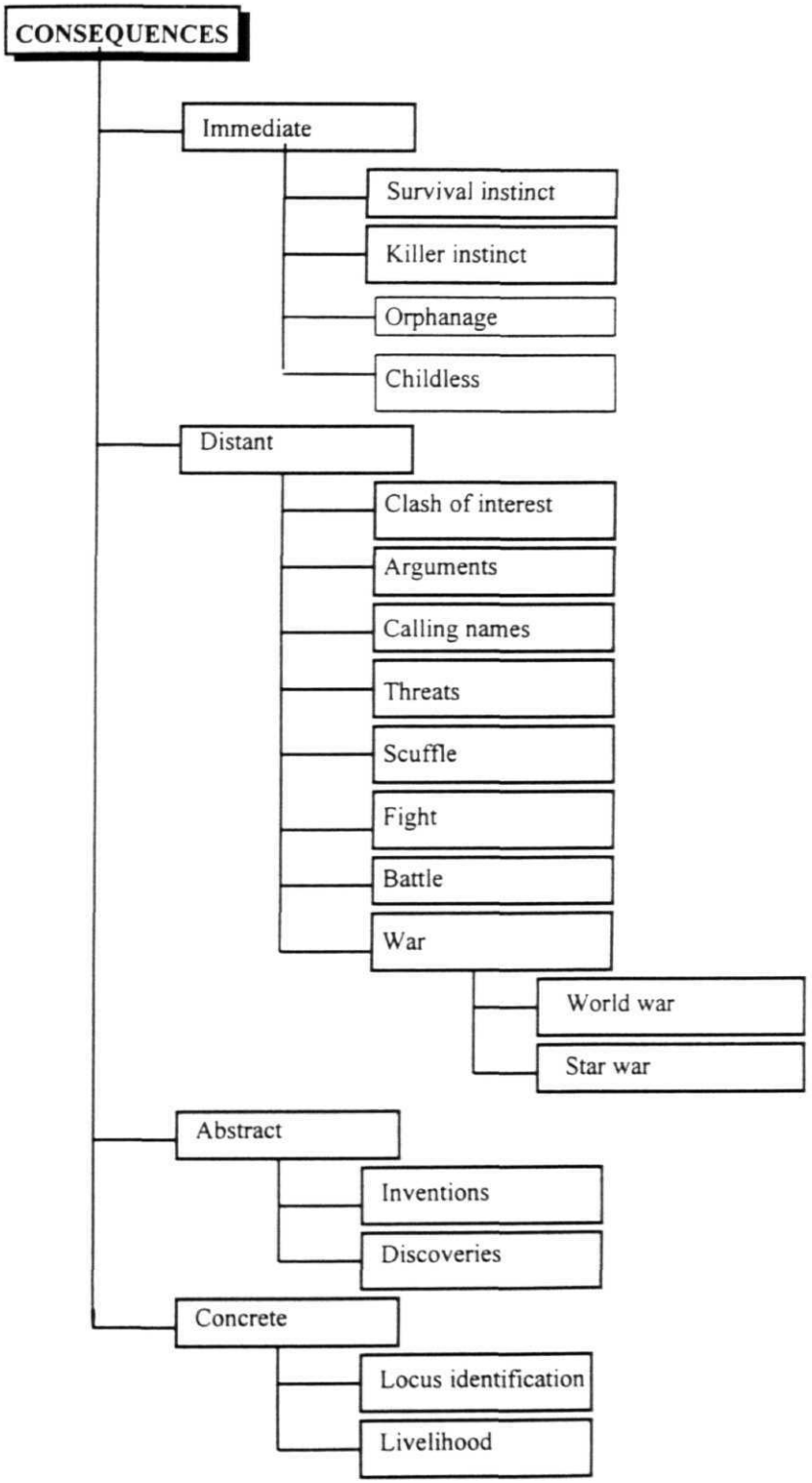
3.2.1. DOMAIN : *SELF*



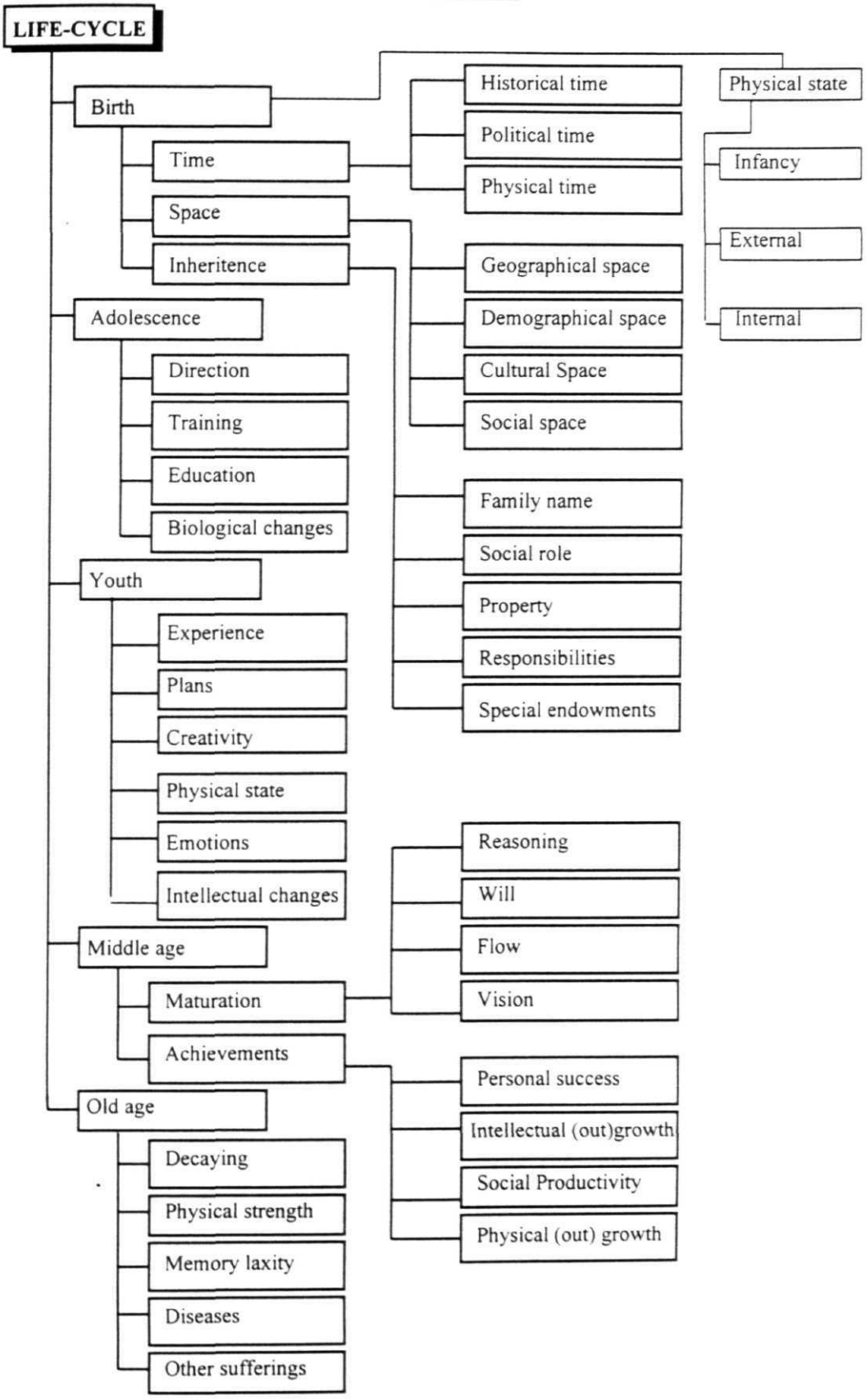
1. DOMAIN : SELF, EXISTENCE,THEORETICAL, ***CONDITIONS***



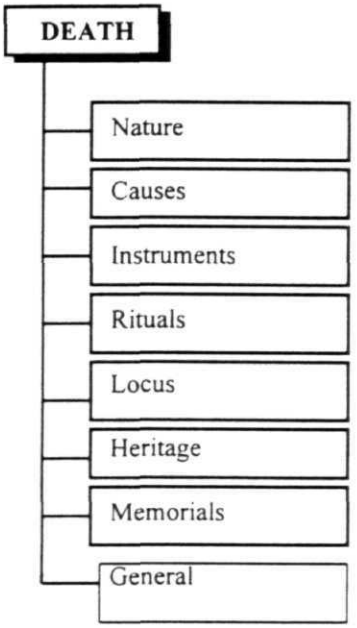
1. DOMAIN : SELF, EXISTENCE,THEORETICAL,***CONSEQUENCES***



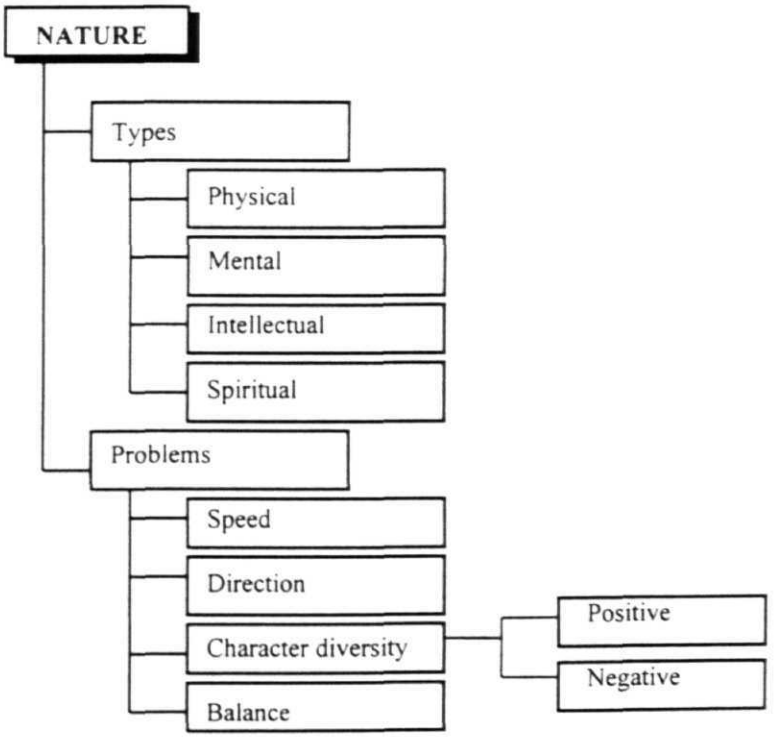
1. DOMAIN : SELF, EXISTENCE,PRACTICAL,***LIFE-CYCLE***



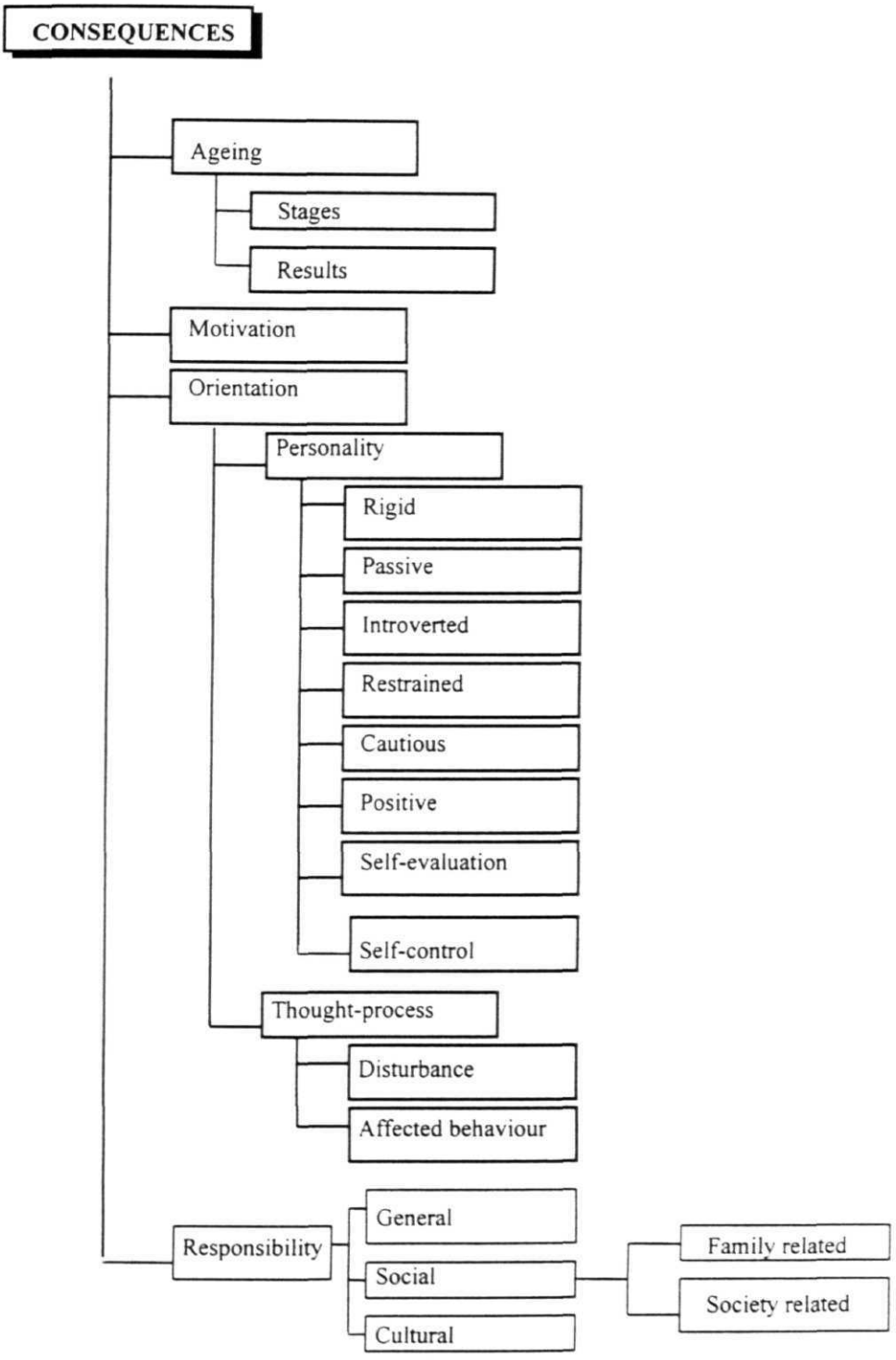
1. DOMAIN : SELF, EXISTENCE,PRACTICAL,***DEATH***



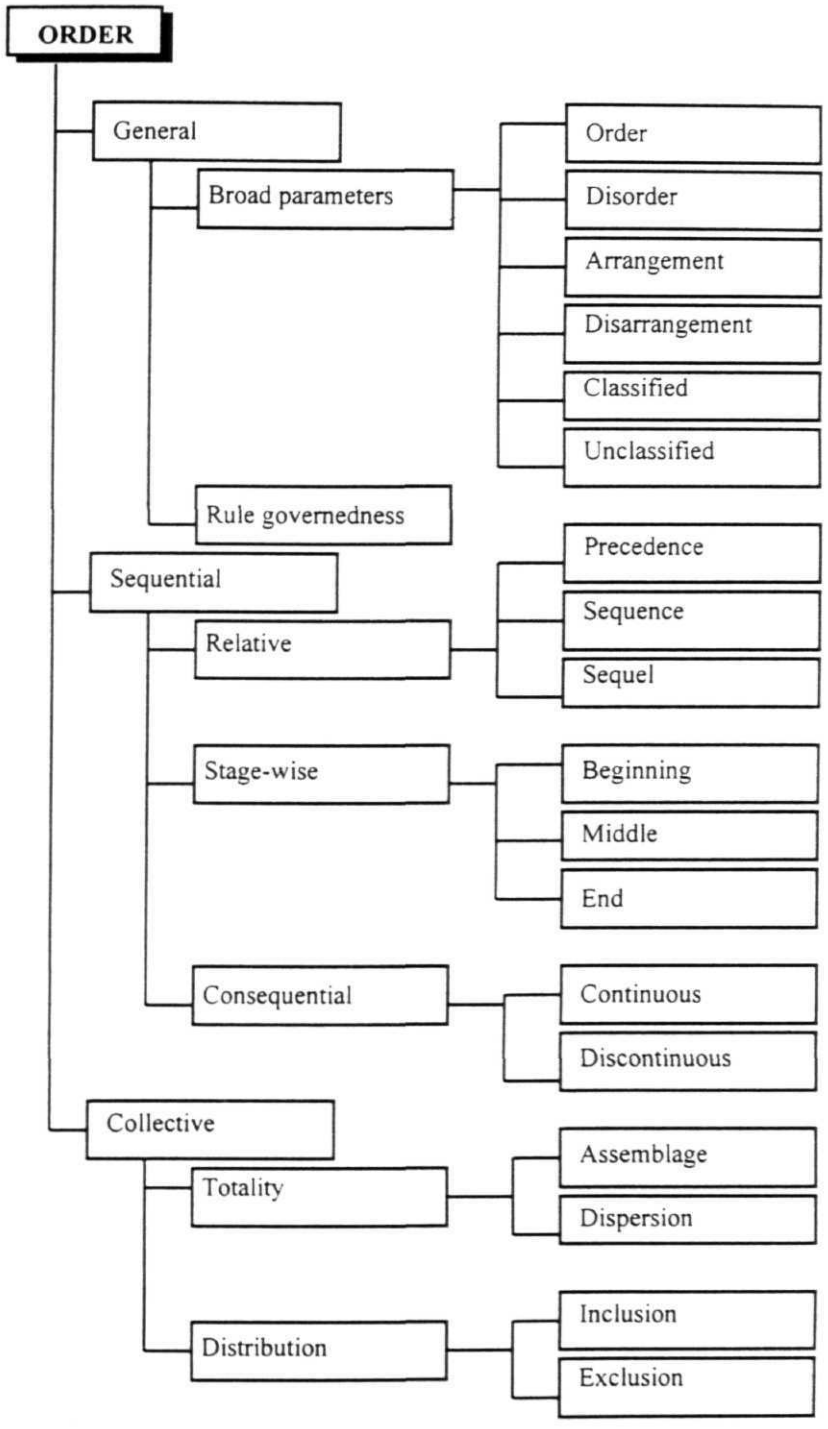
1. DOMAIN : SELF, GROWTH, PROCESSES,***NATURE***



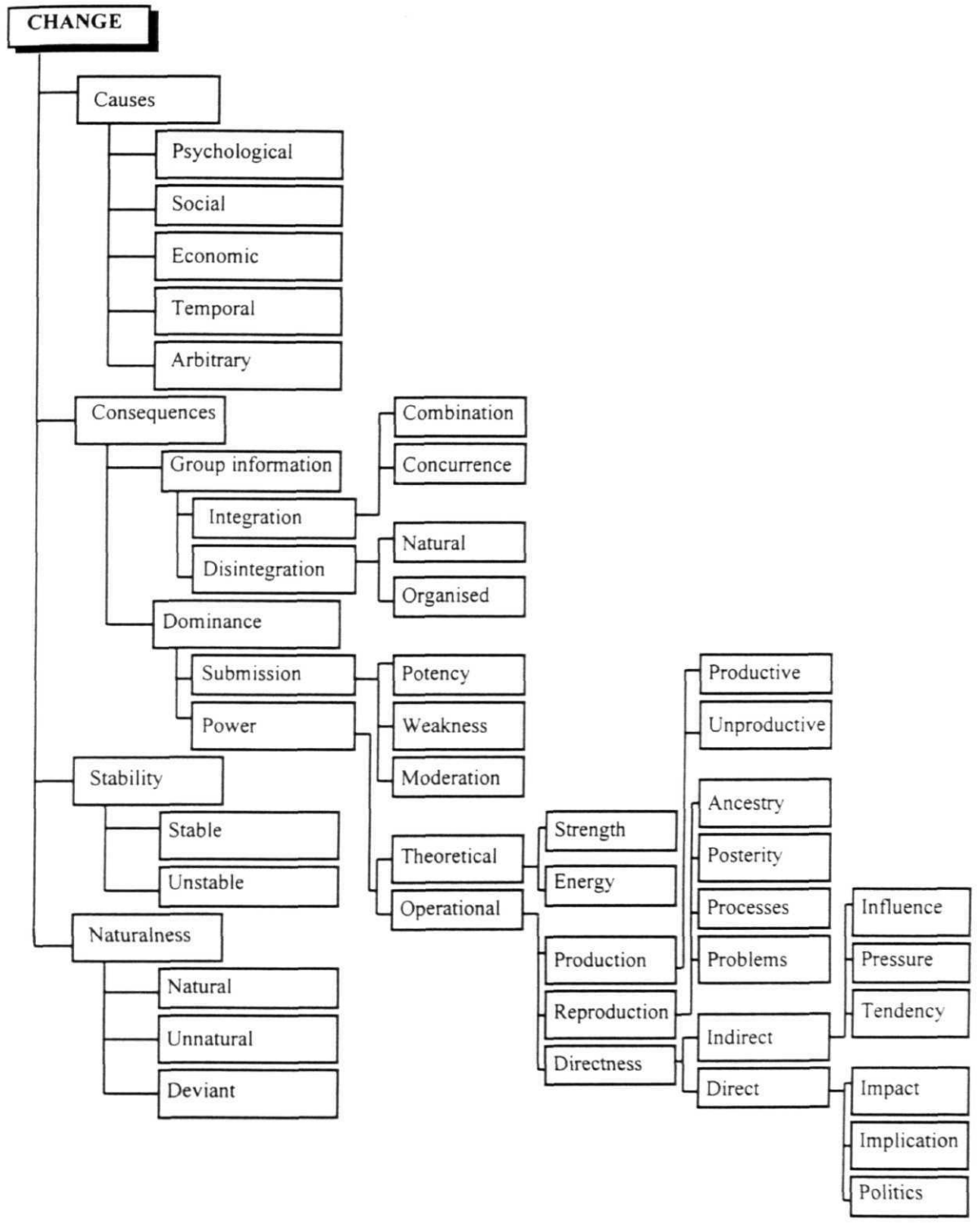
1. DOMAIN : SELF, GROWTH, PROCESSES,***CONSEQUENCES***



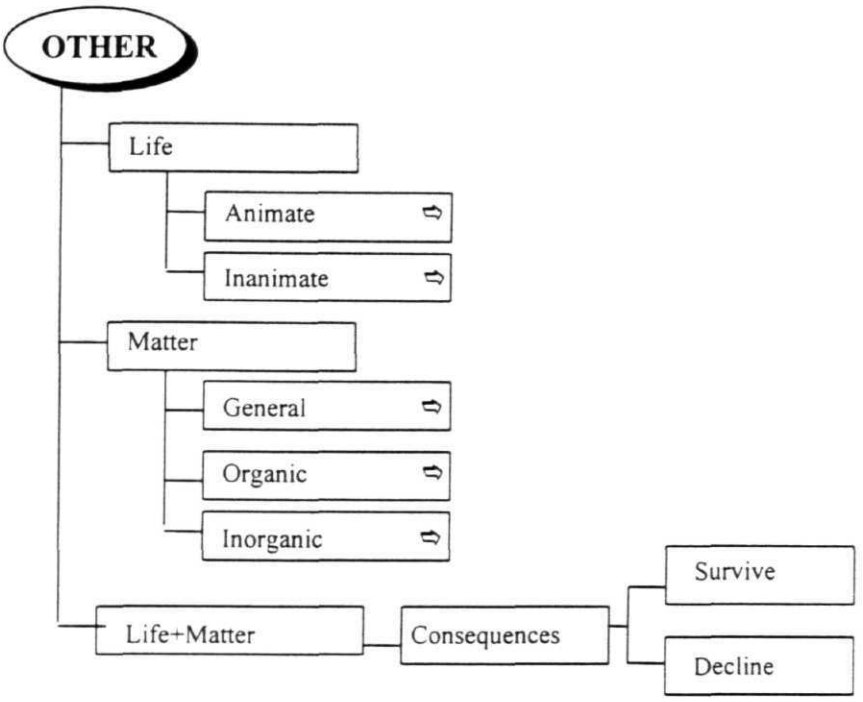
1. DOMAIN : SELF, GROWTH, PRODUCTS,***ORDER***



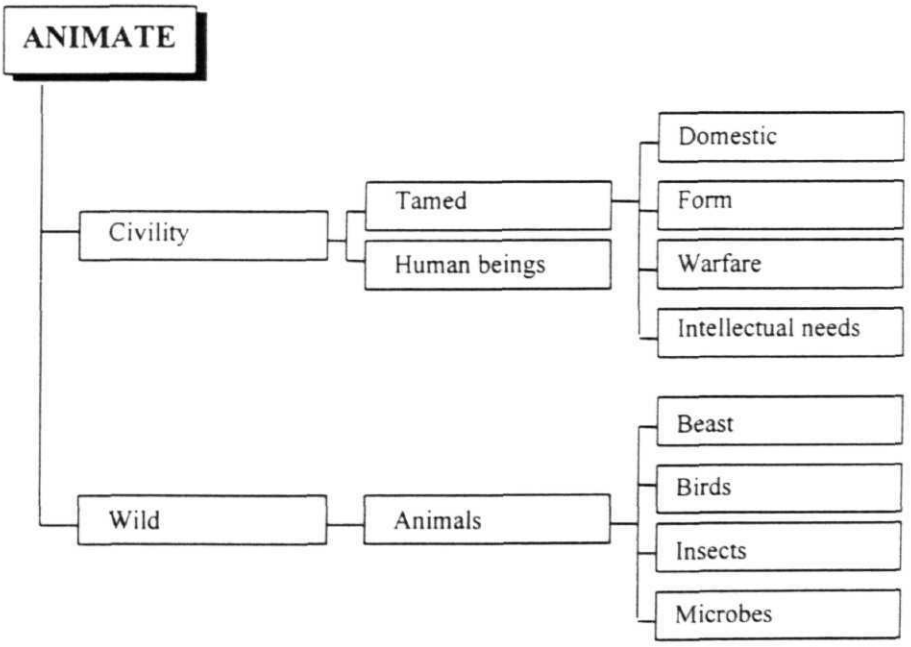
1. DOMAIN : SELF, GROWTH,PRODUCTS, ***CHANGE***



3.2.2. DOMAIN : *OTHER*

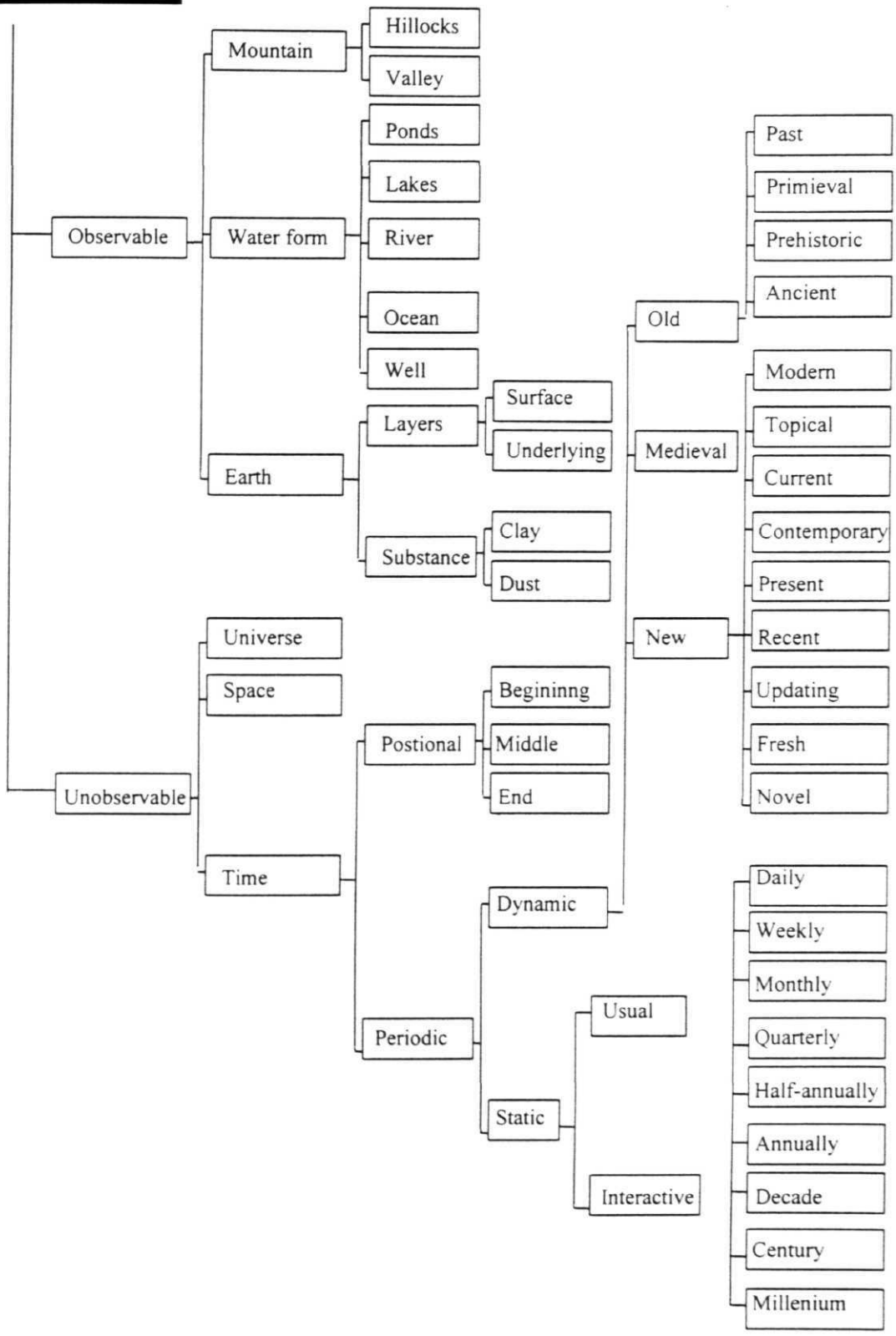


2. DOMAIN : OTHER.LIFE.***ANIMATE***



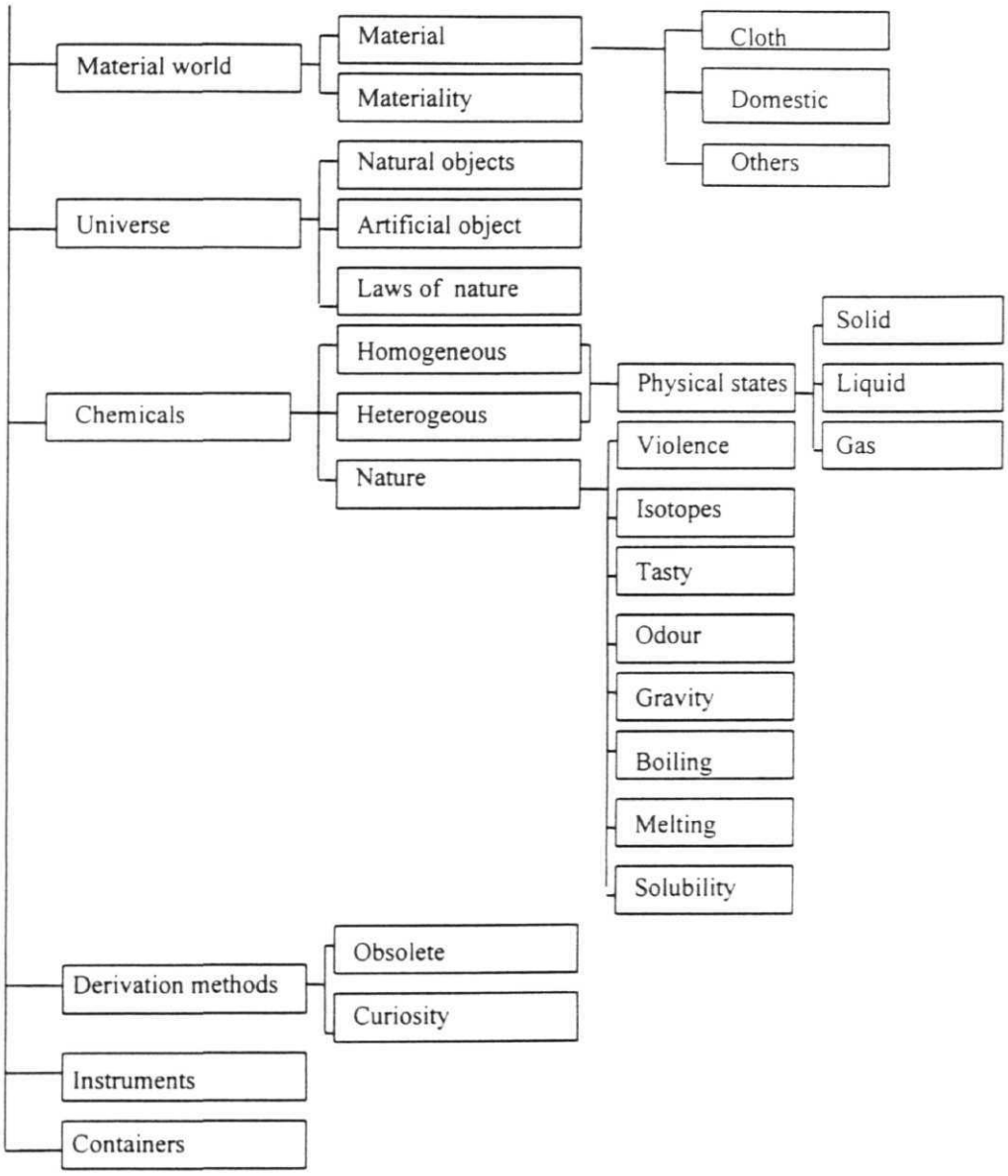
2. DOMAIN : OTHER,LIFE,***INANIMATE***

INANIMATE

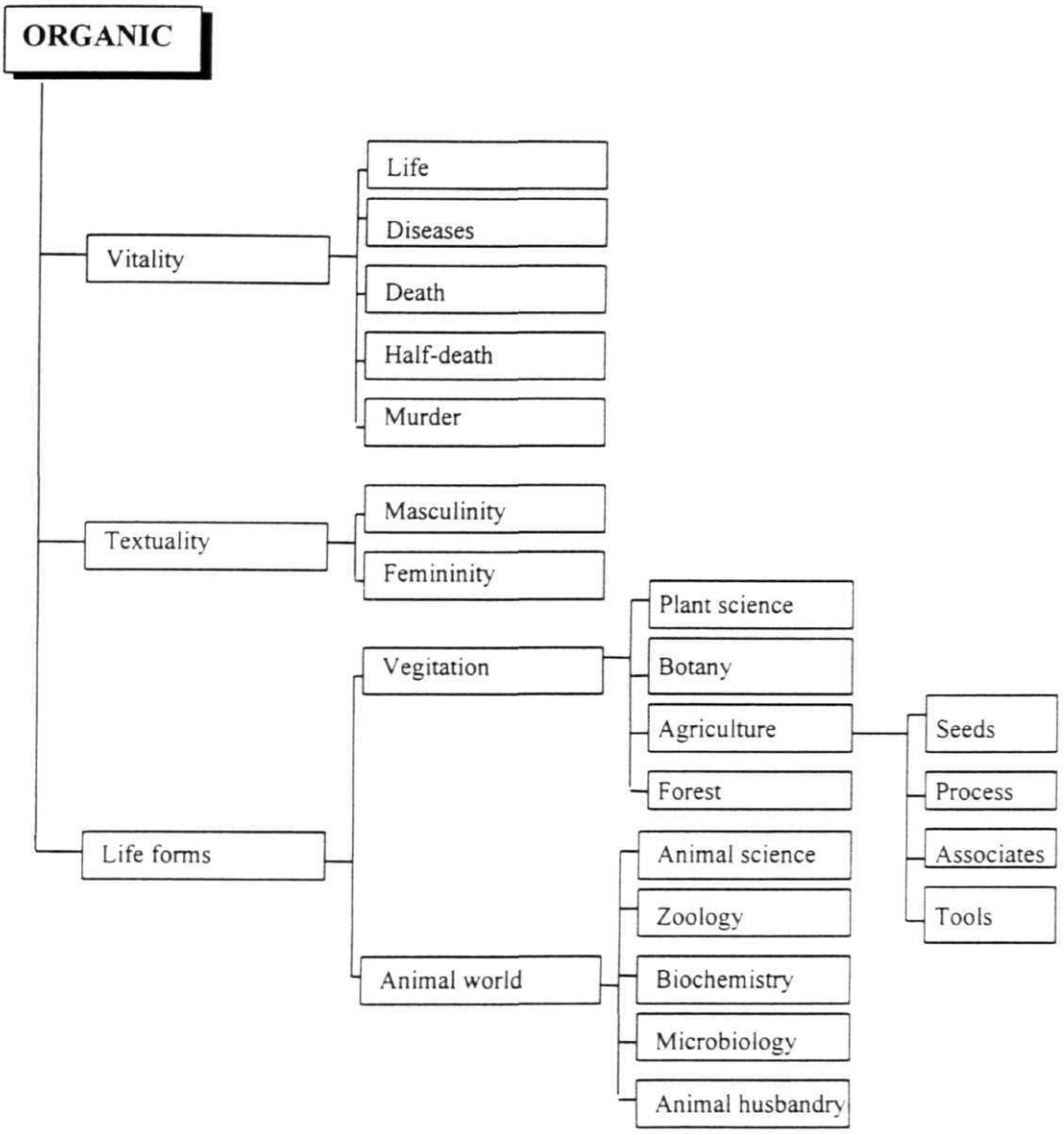


2. DOMAIN : OTHER,MATTER,***GENERAL***

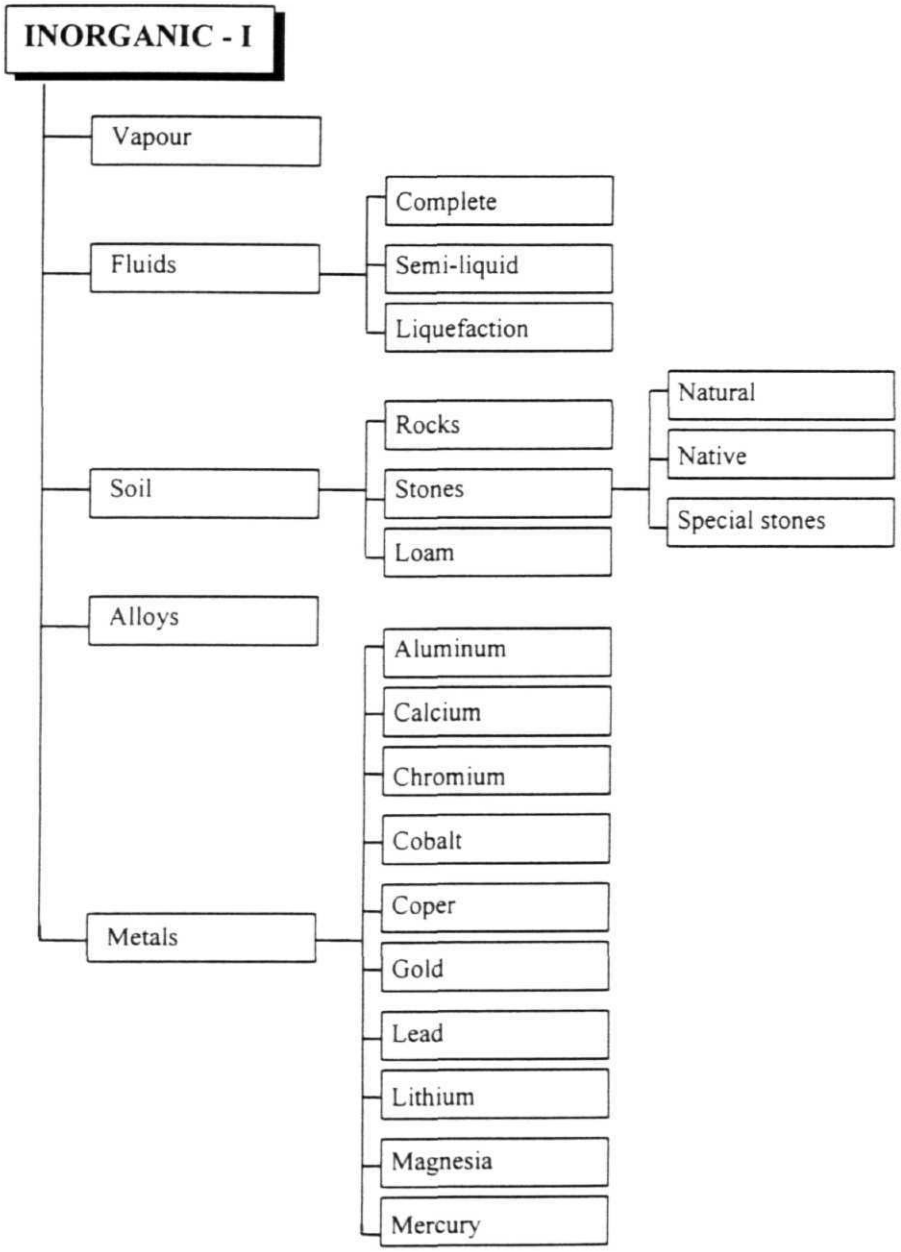
GENERAL



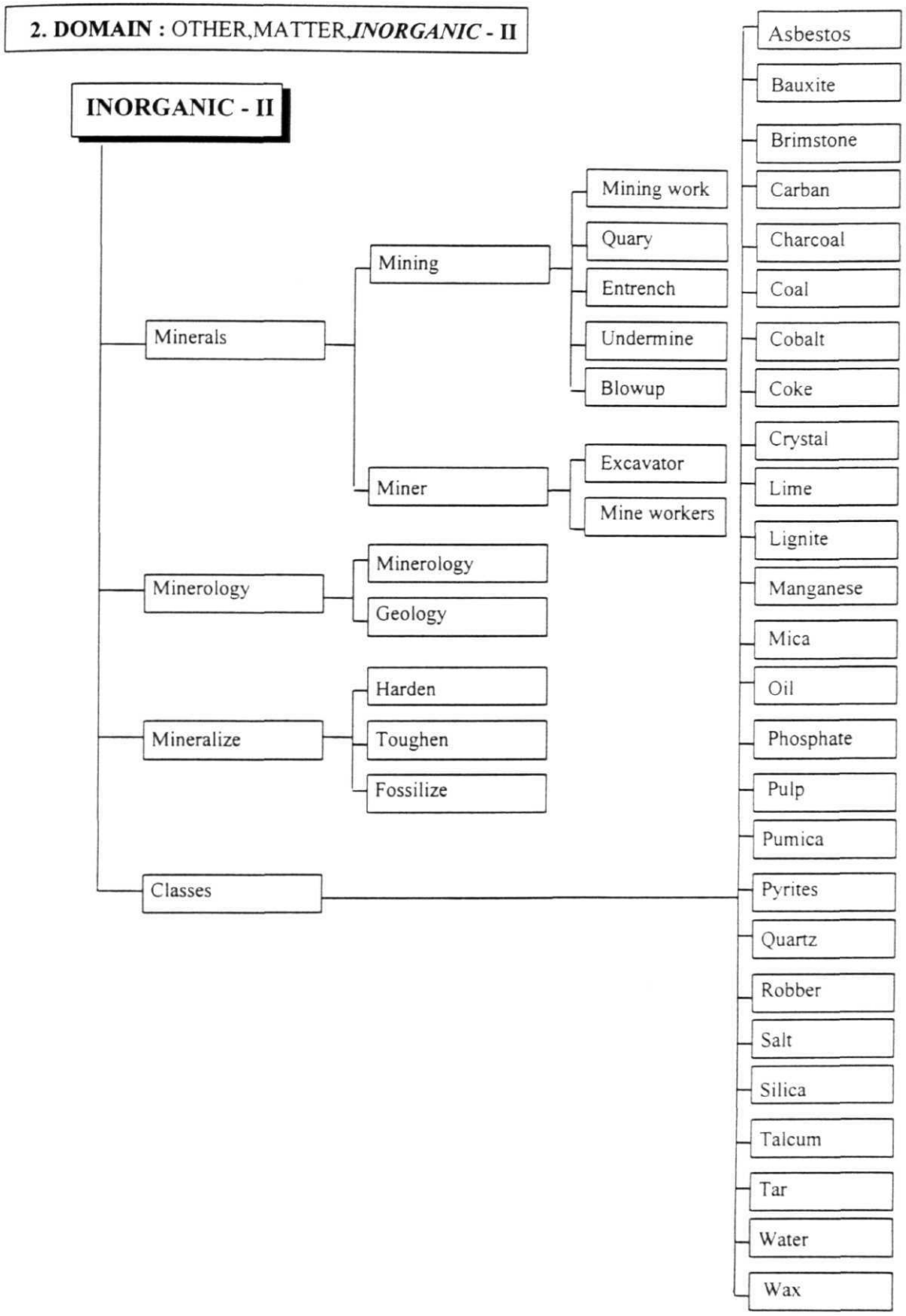
2. DOMAIN : OTHER,MATTER,***ORGANIC***



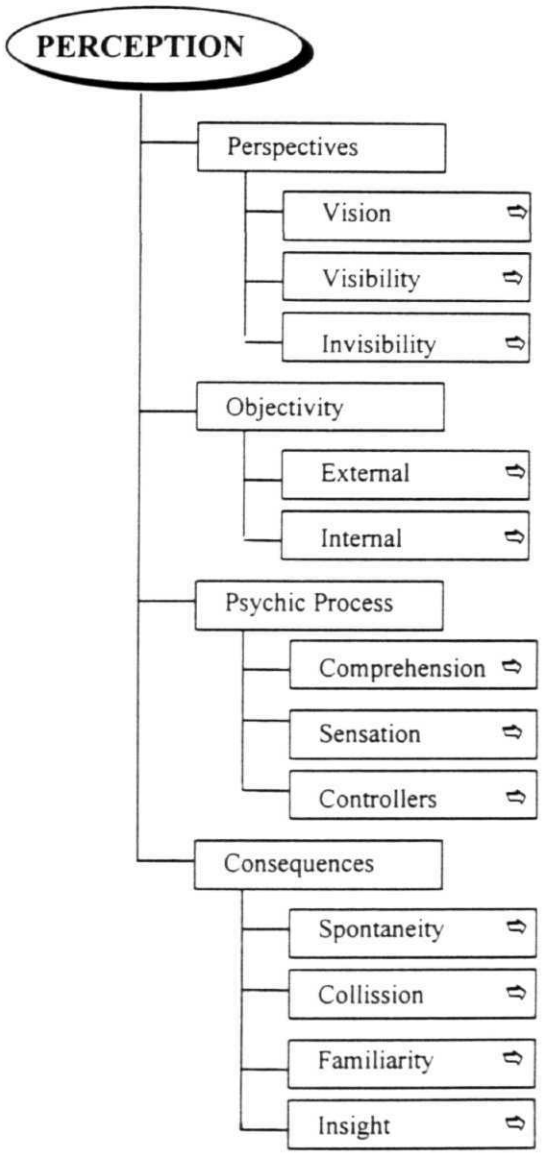
2. DOMAIN : OTHER,MATTER,***INORGANIC*** - **I**



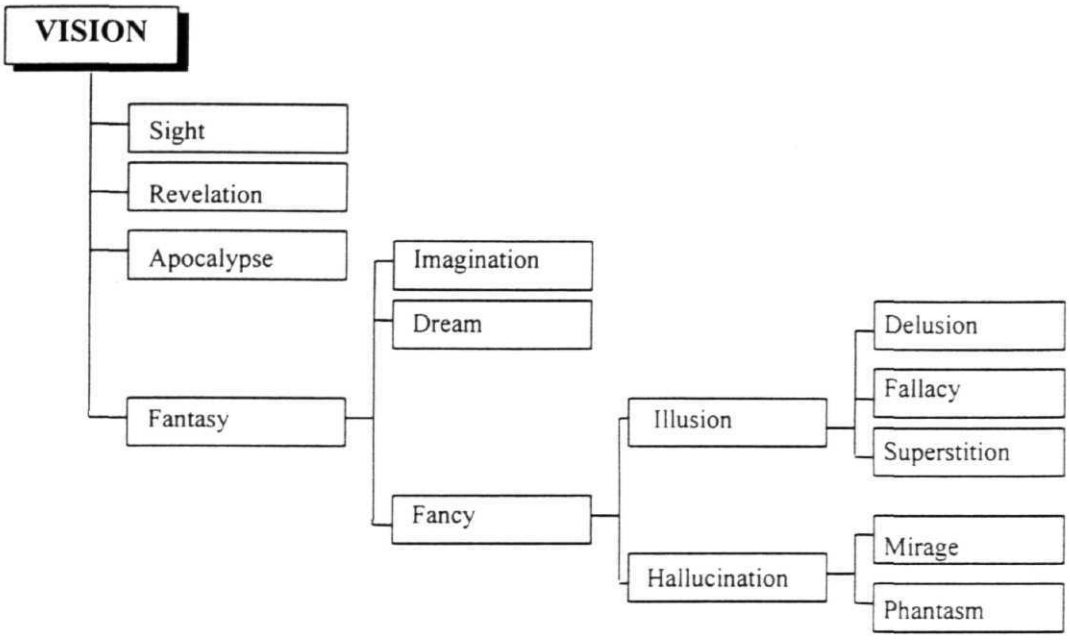
2. DOMAIN : OTHER,MATTER,***INORGANIC*** **- II**



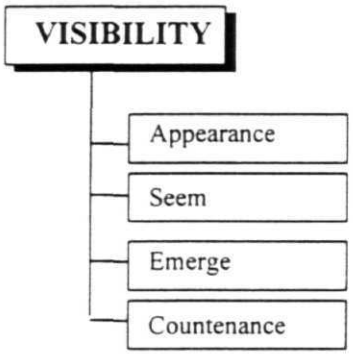
3.3.3. DOMAIN : *PERCEPTION*



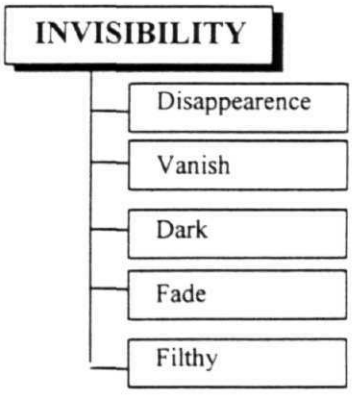
3. DOMAIN : PERCEPTION,PERSPECTIVES,***VISION***



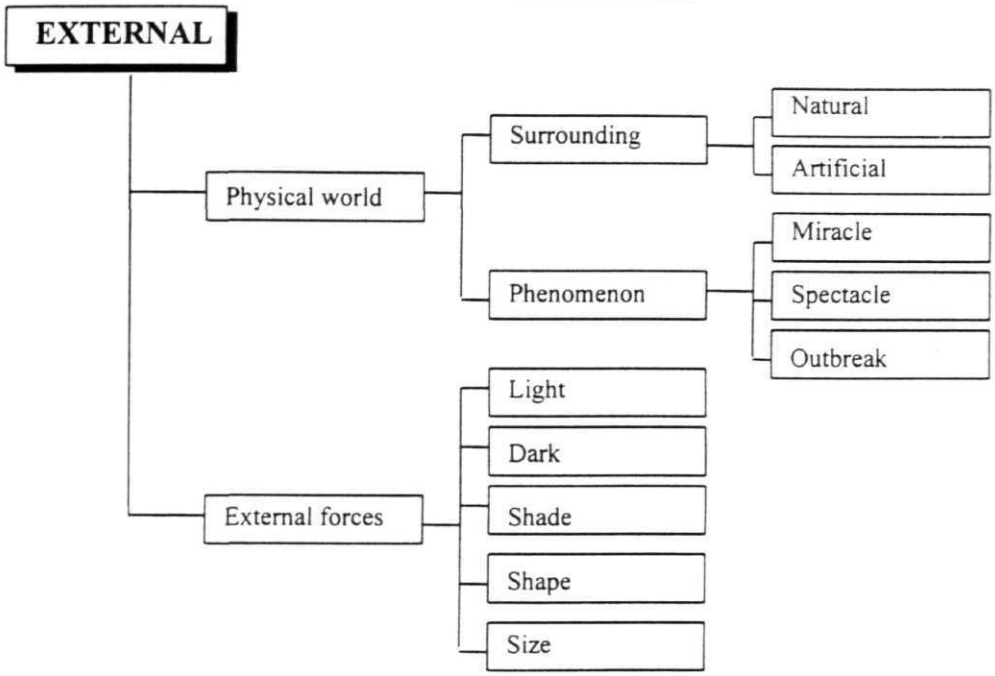
3. DOMAIN : PERCEPTION,PERSPECTIVES,***VISIBILITY***



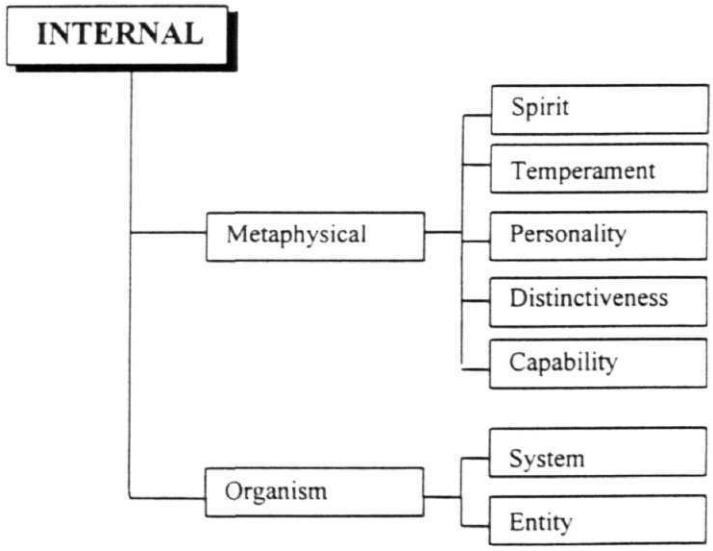
3. DOMAIN : PERCEPTION,PERSPECTIVES,***INVISIBILITY***



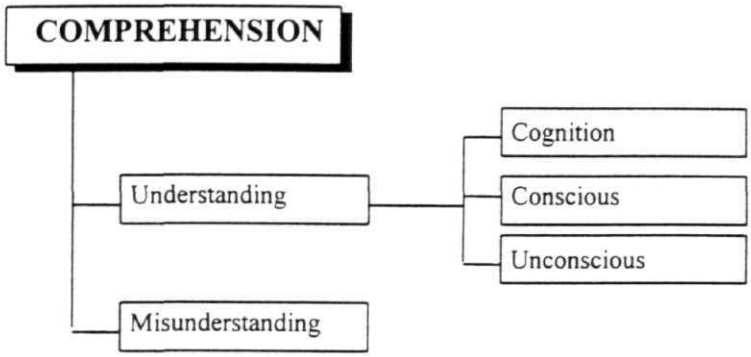
3. DOMAIN : PERCEPTION,OBJECTIVITY,***EXTERNAL***



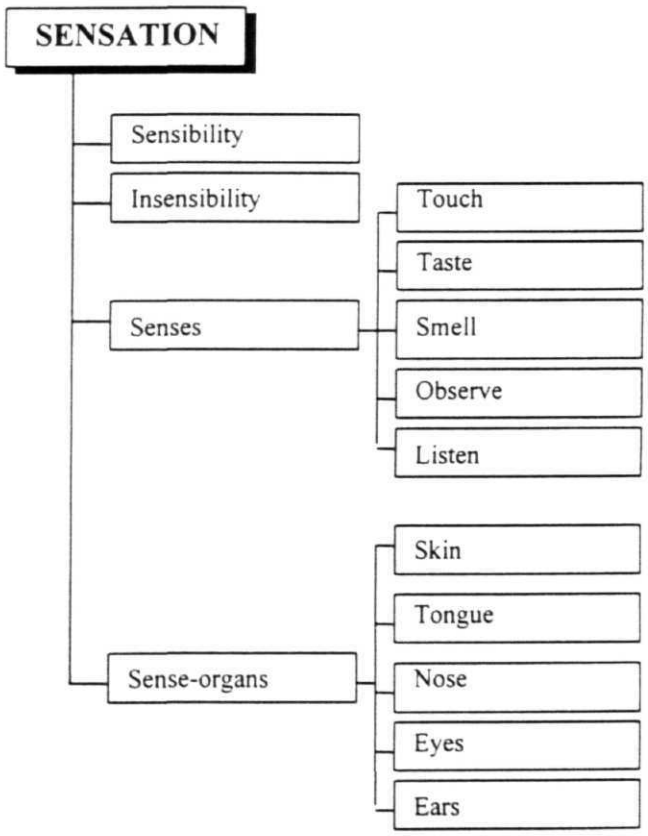
3. DOMAIN : PERCEPTION,OBJECTIVITY,***INTERNAL***



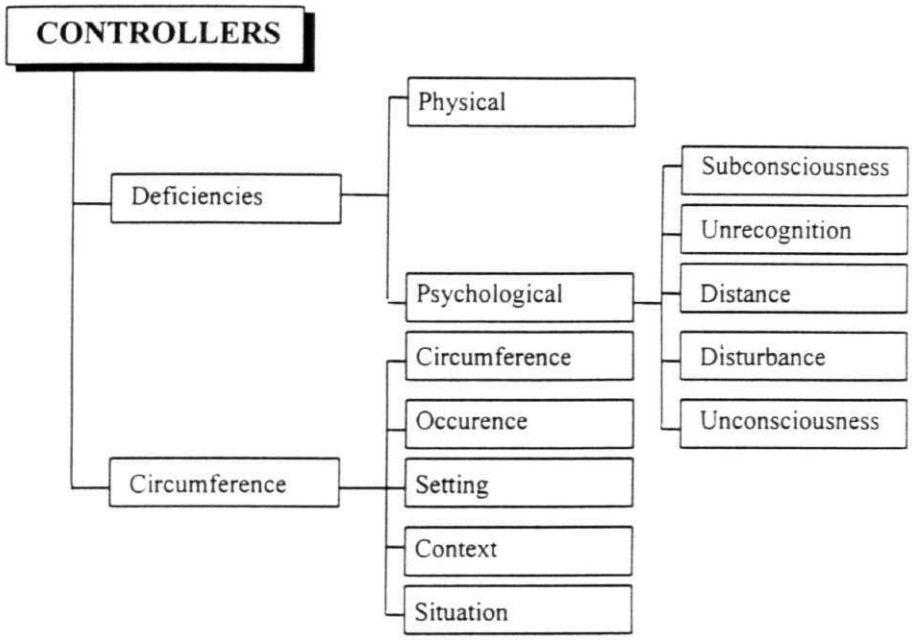
3. DOMAIN : PERCEPTION,PSYCHIC PROCESS,***COMPREHENSION***



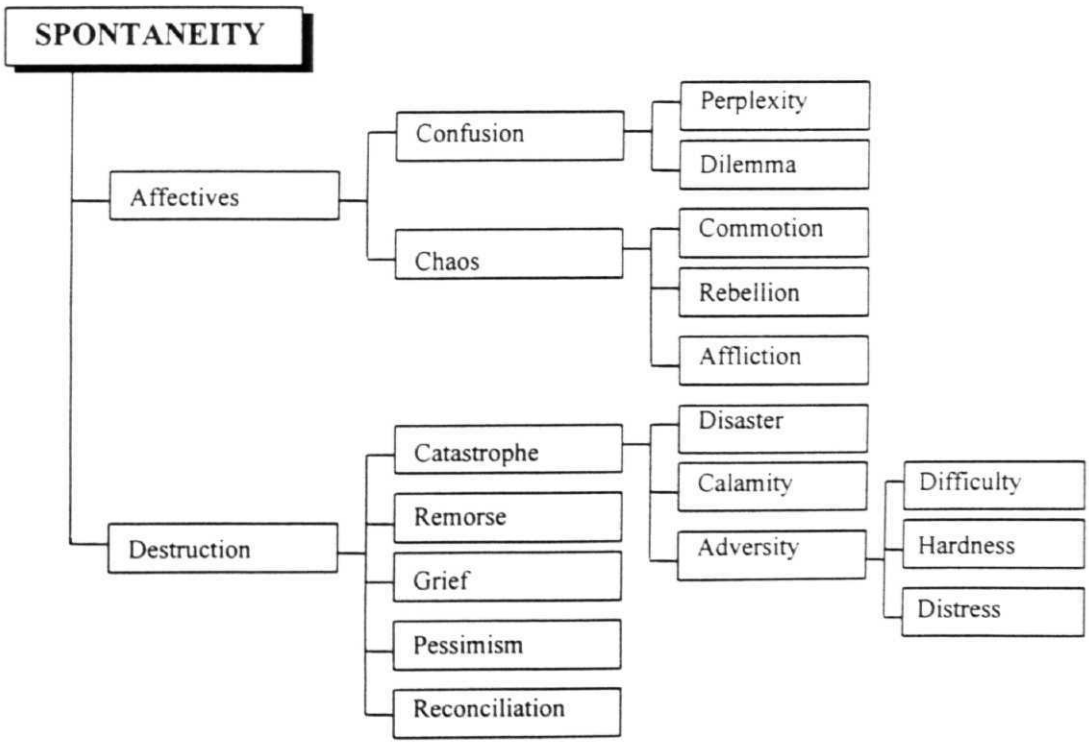
3. DOMAIN : PERCEPTION,PSYCHIC PROCESS,***SENSATION***



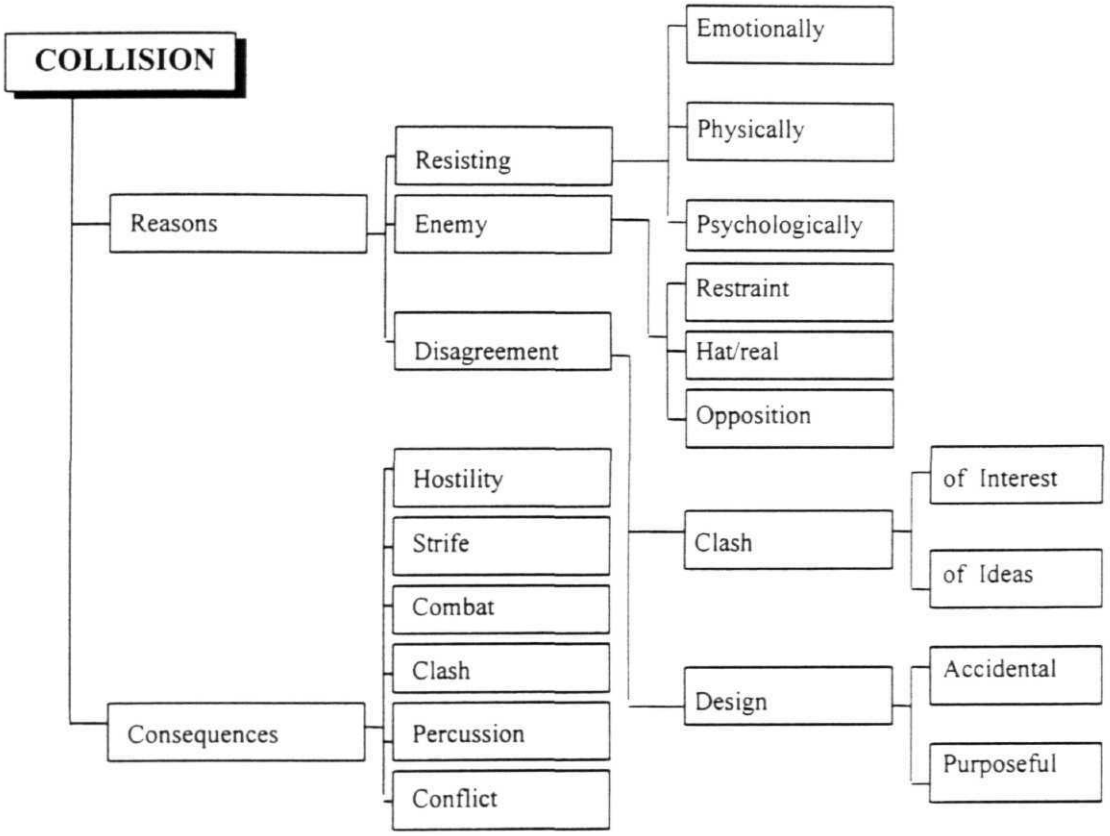
3. DOMAIN : PERCEPTION,PSYCHIC PROCESS,***CONTROLLERS***



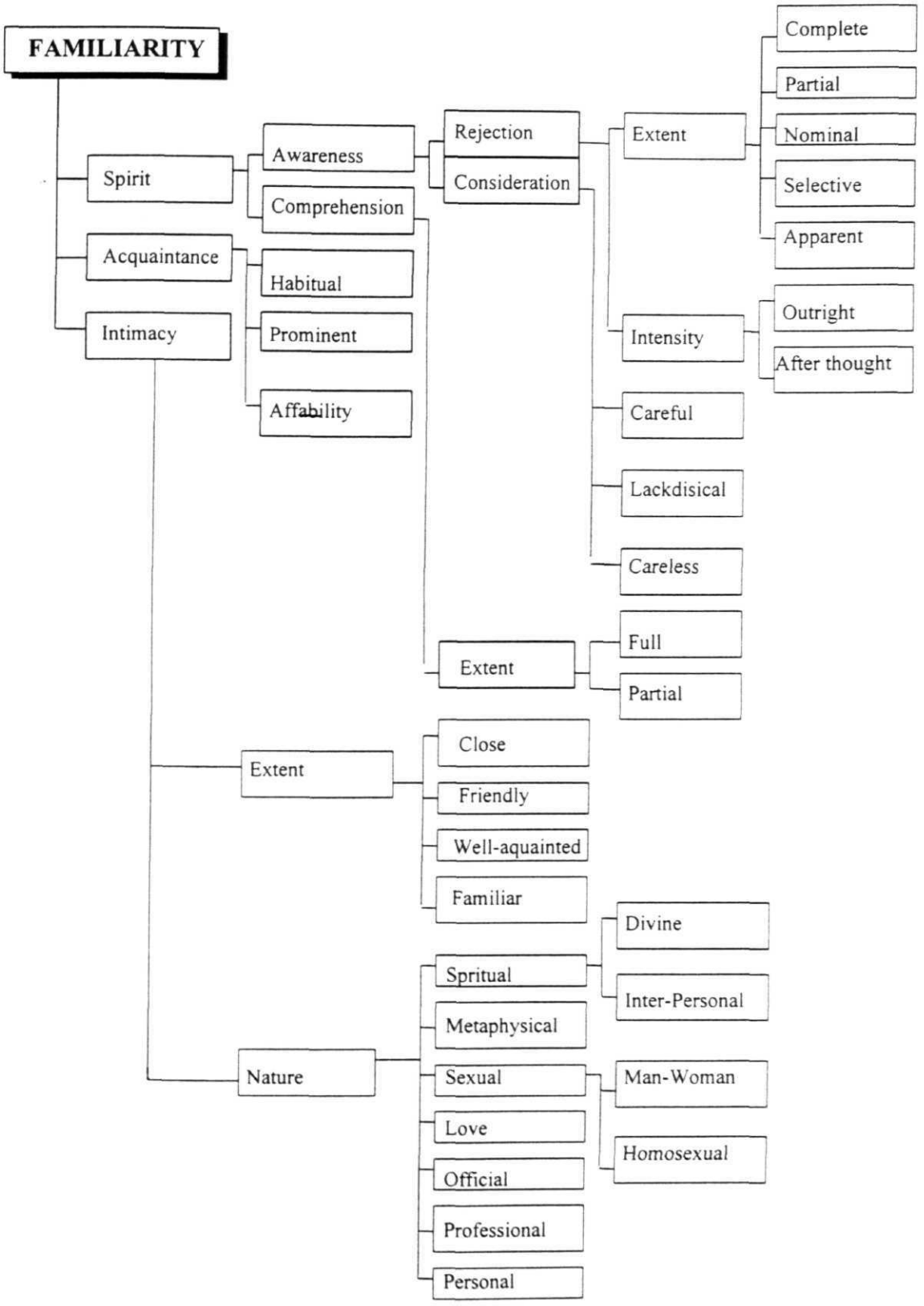
3. DOMAIN : PERCEPTION,CONSEQUENCES,***SPONTANEITY***



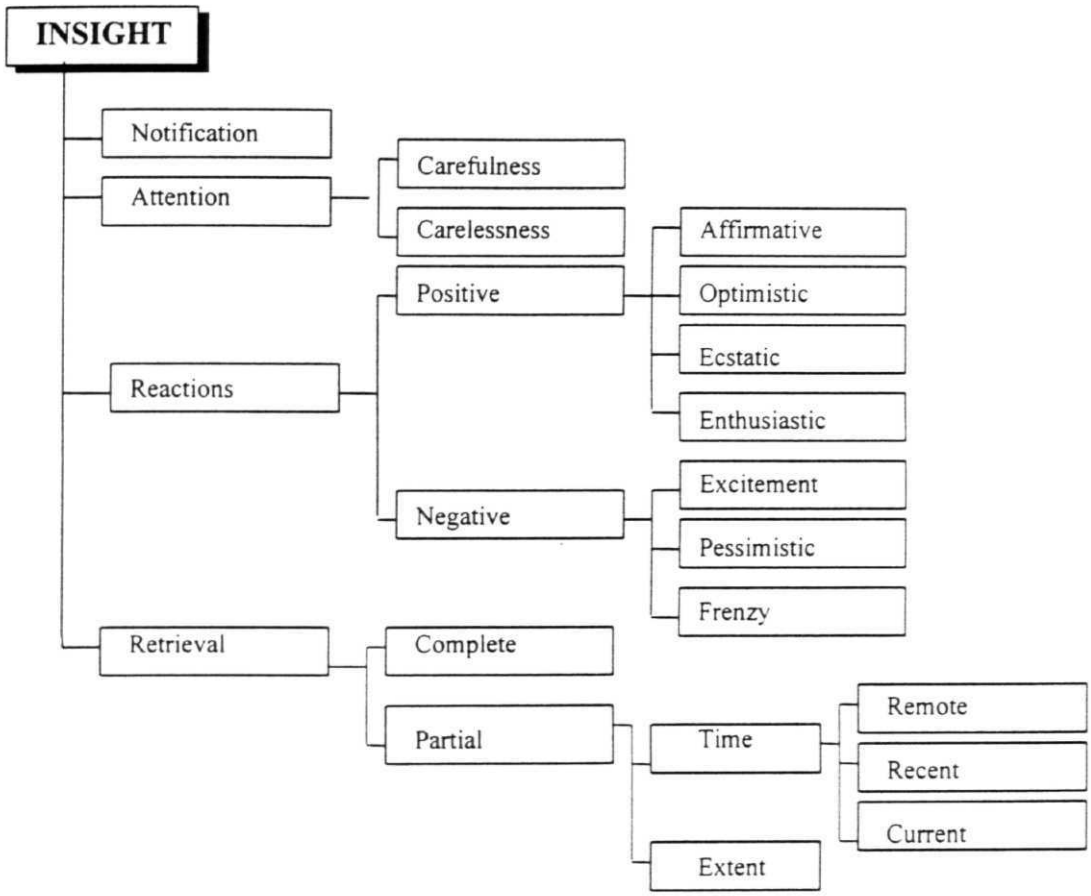
3. DOMAIN : PERCEPTION,CONSEQUENCES,***COLLISION***



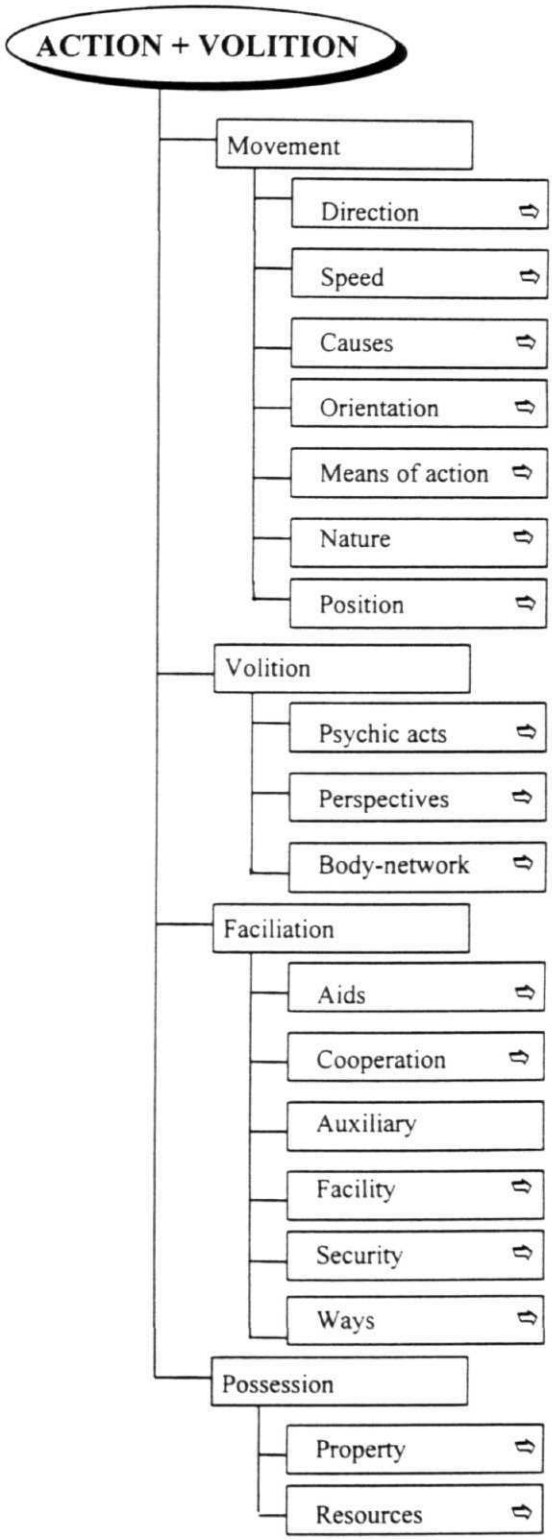
3. DOMAIN : PERCEPTION,CONSEQUENCES,***FAMILIARITY***



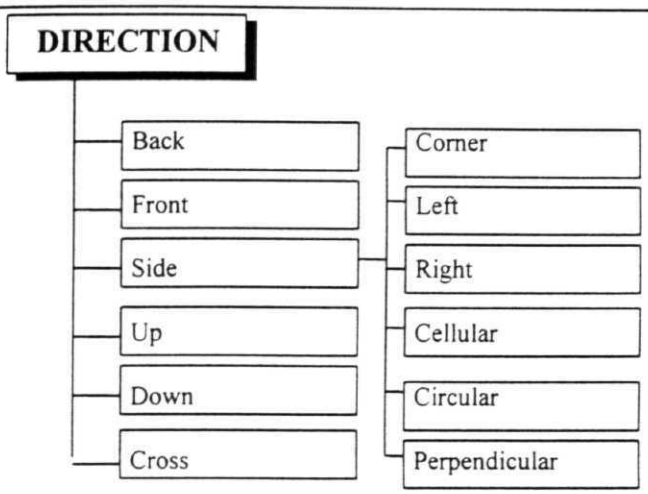
3. DOMAIN : PERCEPTION,CONSEQUENCES,***INSIGHT***



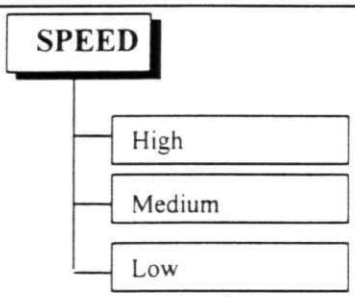
3.2.4. DOMAIN : *ACTION + VOLITION*



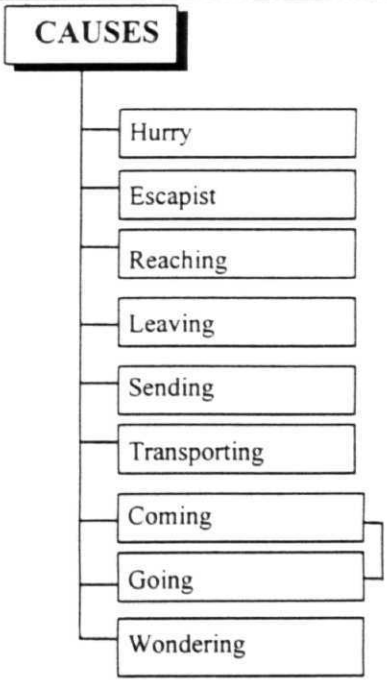
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***DIRECTION***



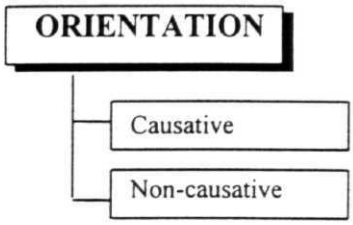
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***SPEED***



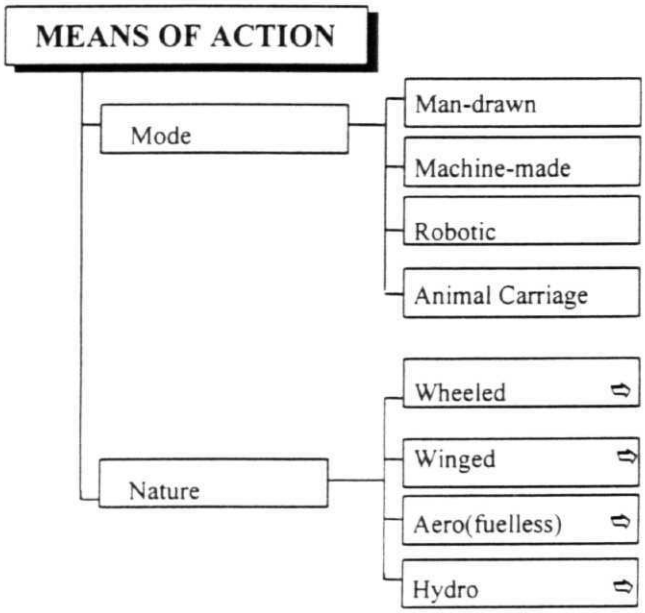
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***CAUSES***



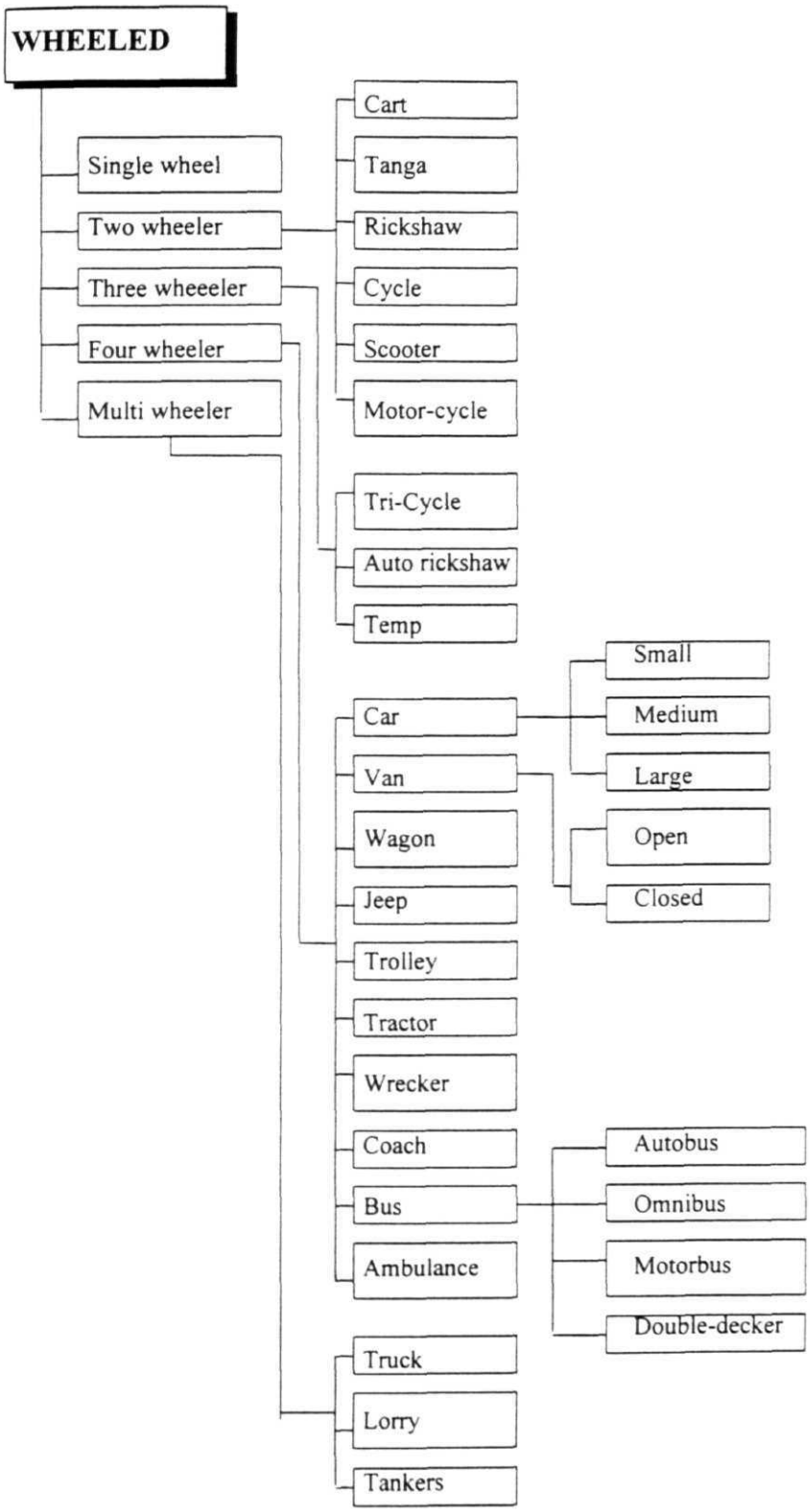
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***ORIENTATION***



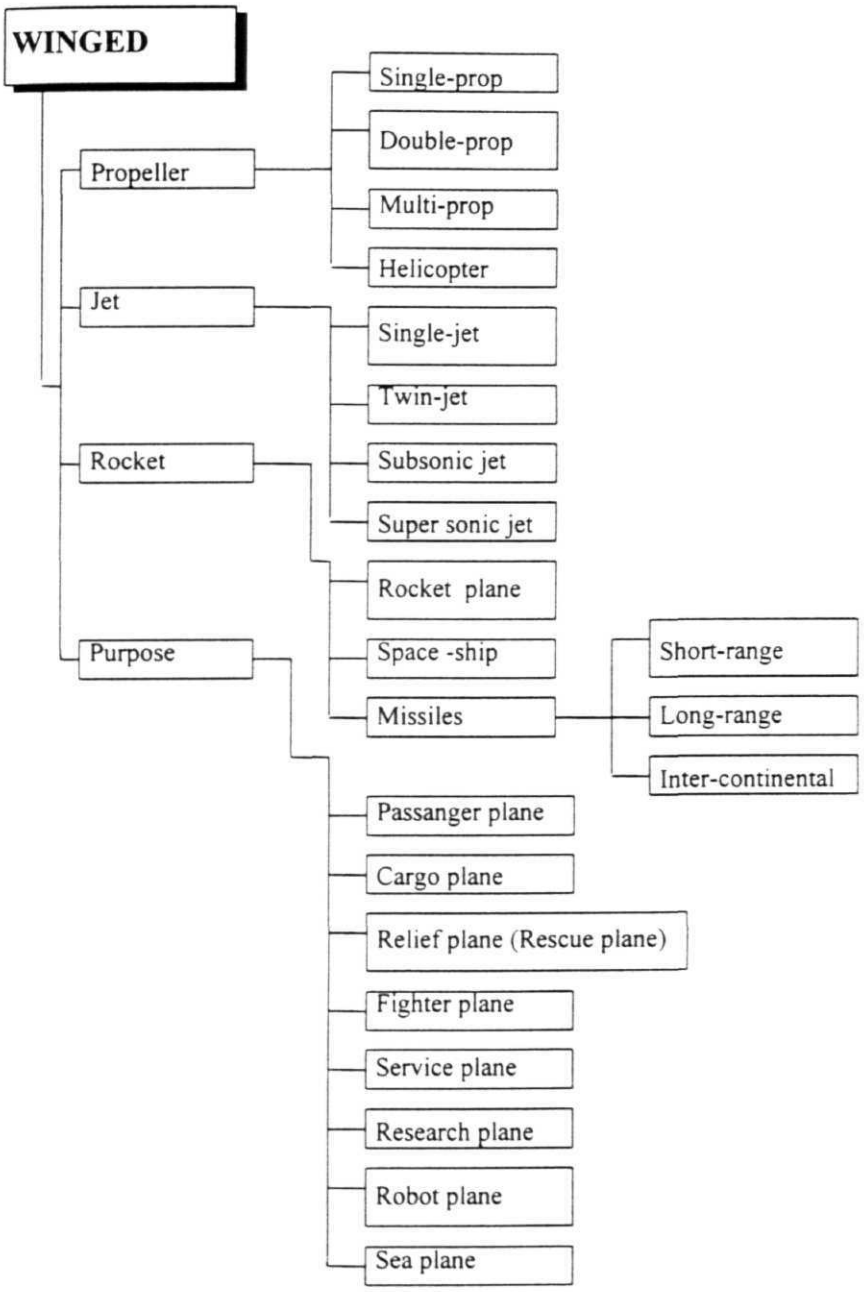
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***MEANS OF ACTION***



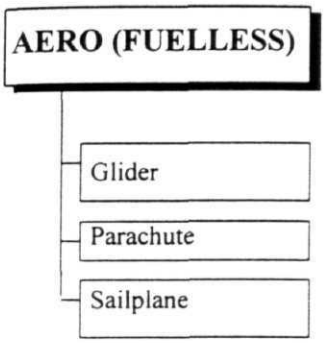
4. DOMAIN : ACTION + VOLITION,MOVEMENT,MEANS OF ACTION, ***WHEELED***



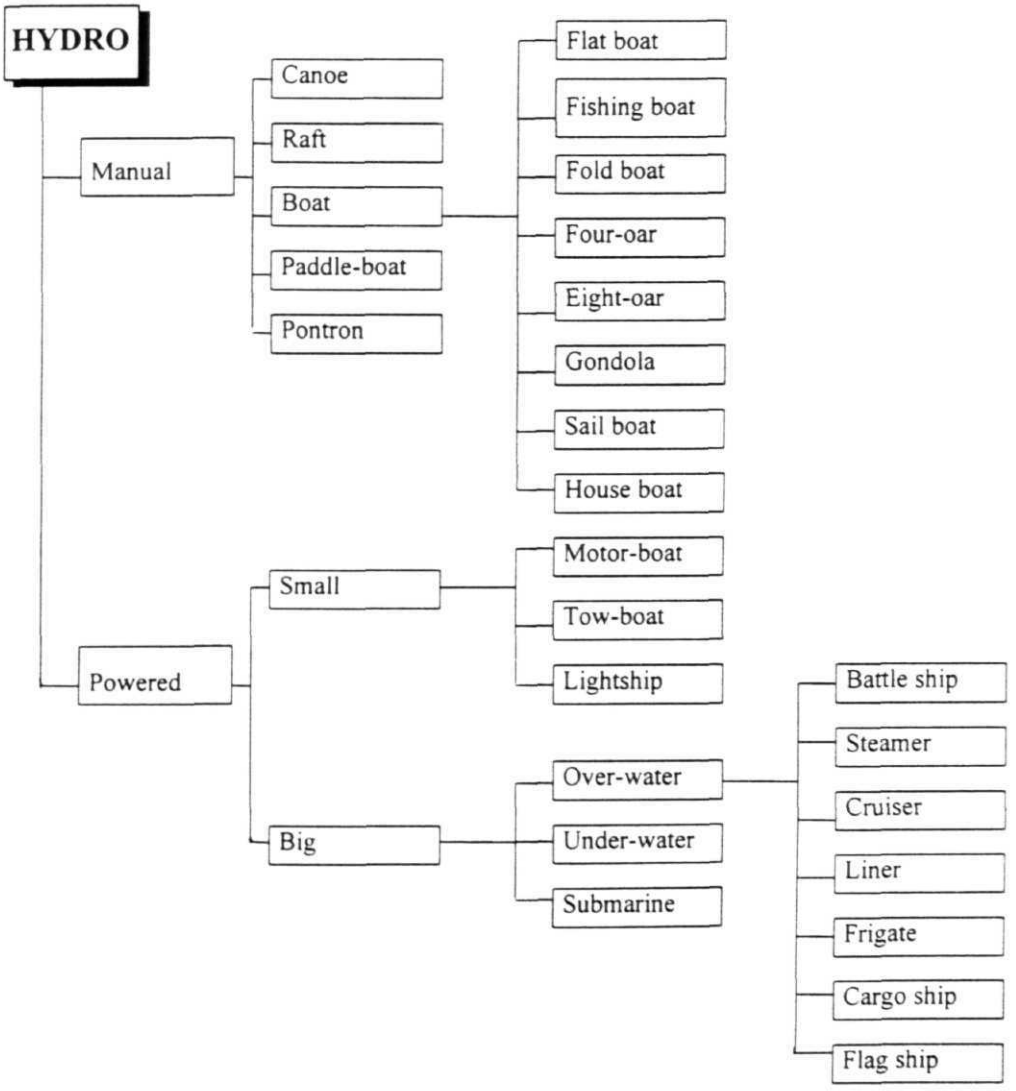
4. DOMAIN : ACTION + VOLITION,MOVEMENT,MEANS OF ACTION, ***WINGED***



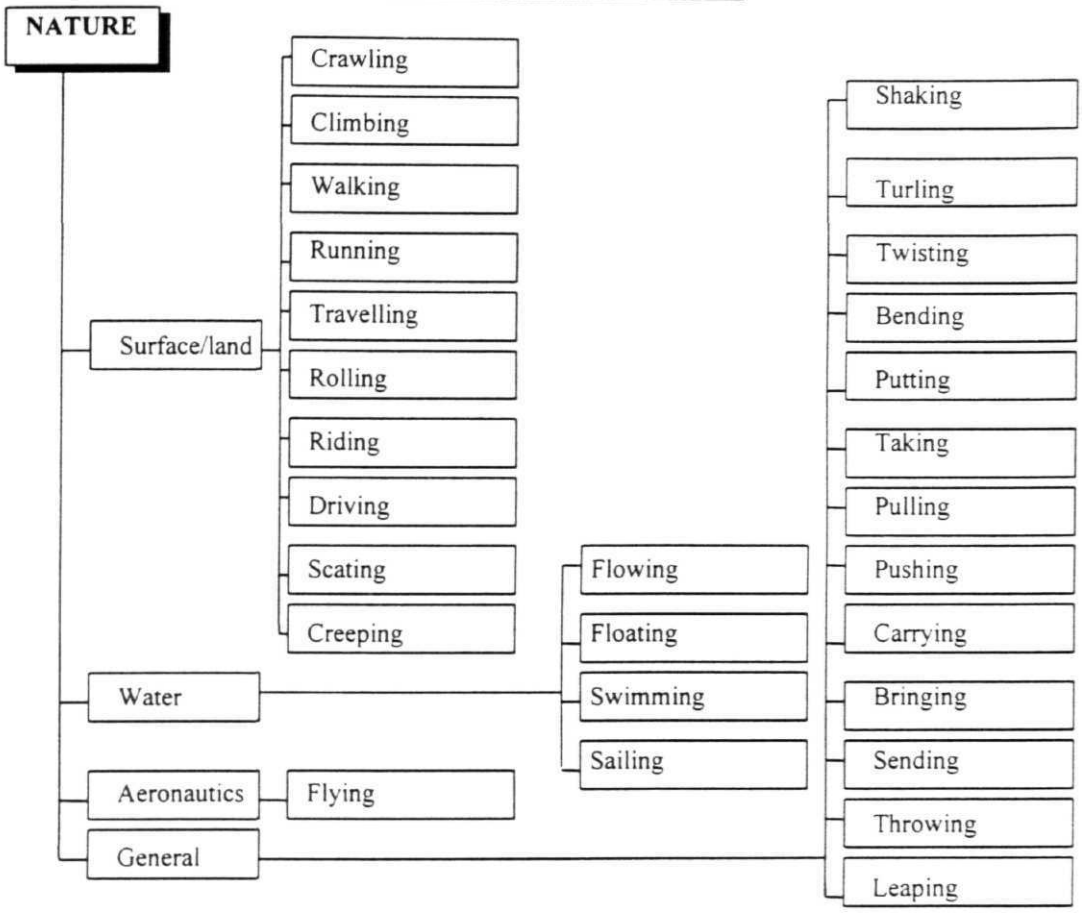
4. DOMAIN : ACTION + VOLITION,MOVEMENT,MEANS OF ACTION, ***AERO (FUELLESS)***



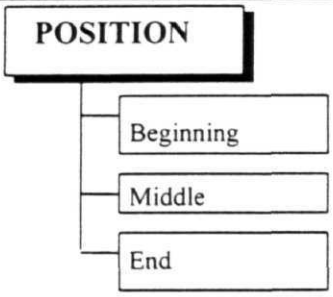
4. DOMAIN : ACTION + VOLITION,MOVEMENT,MEANS OF ACTION, ***HYDRO***



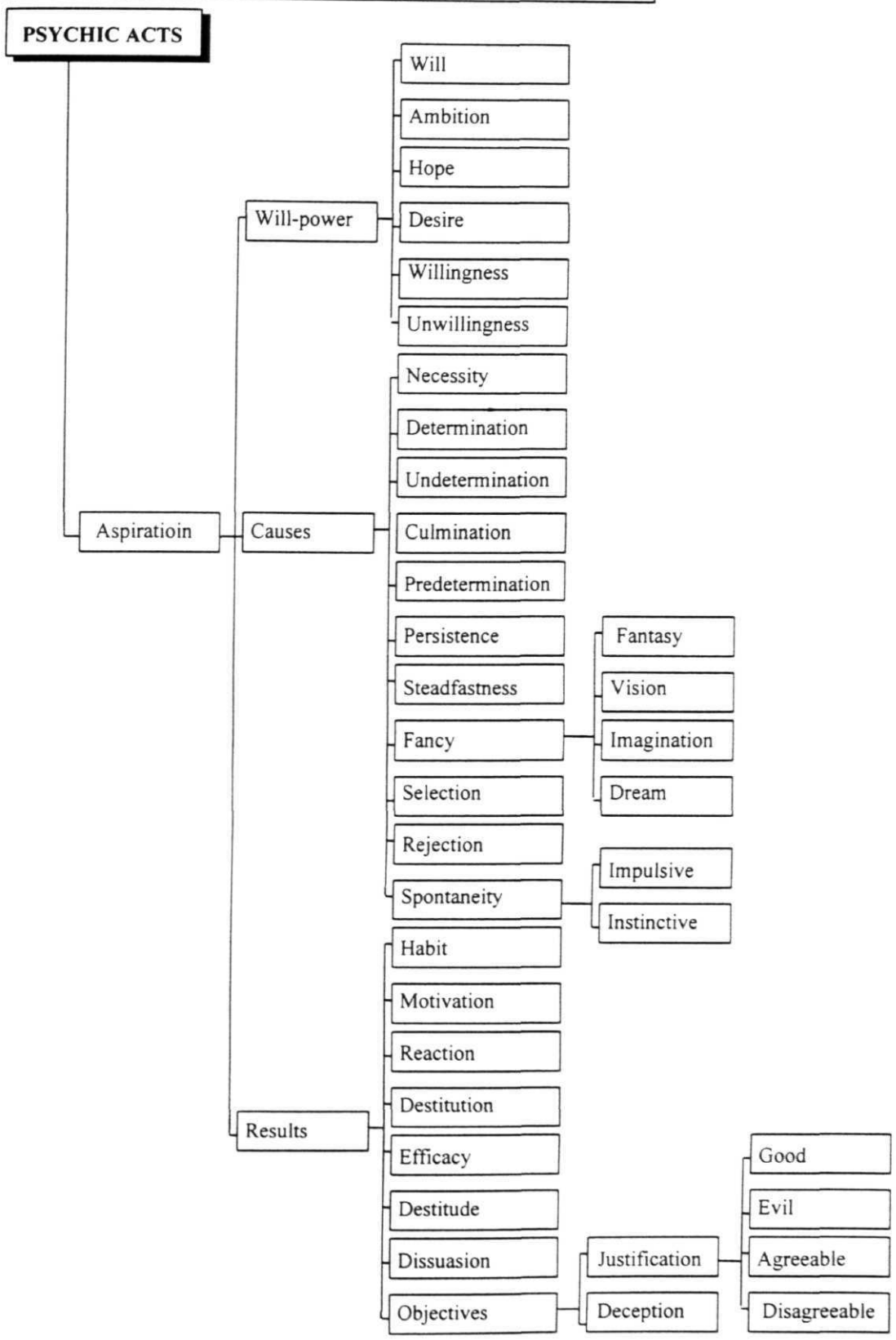
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***NATURE***



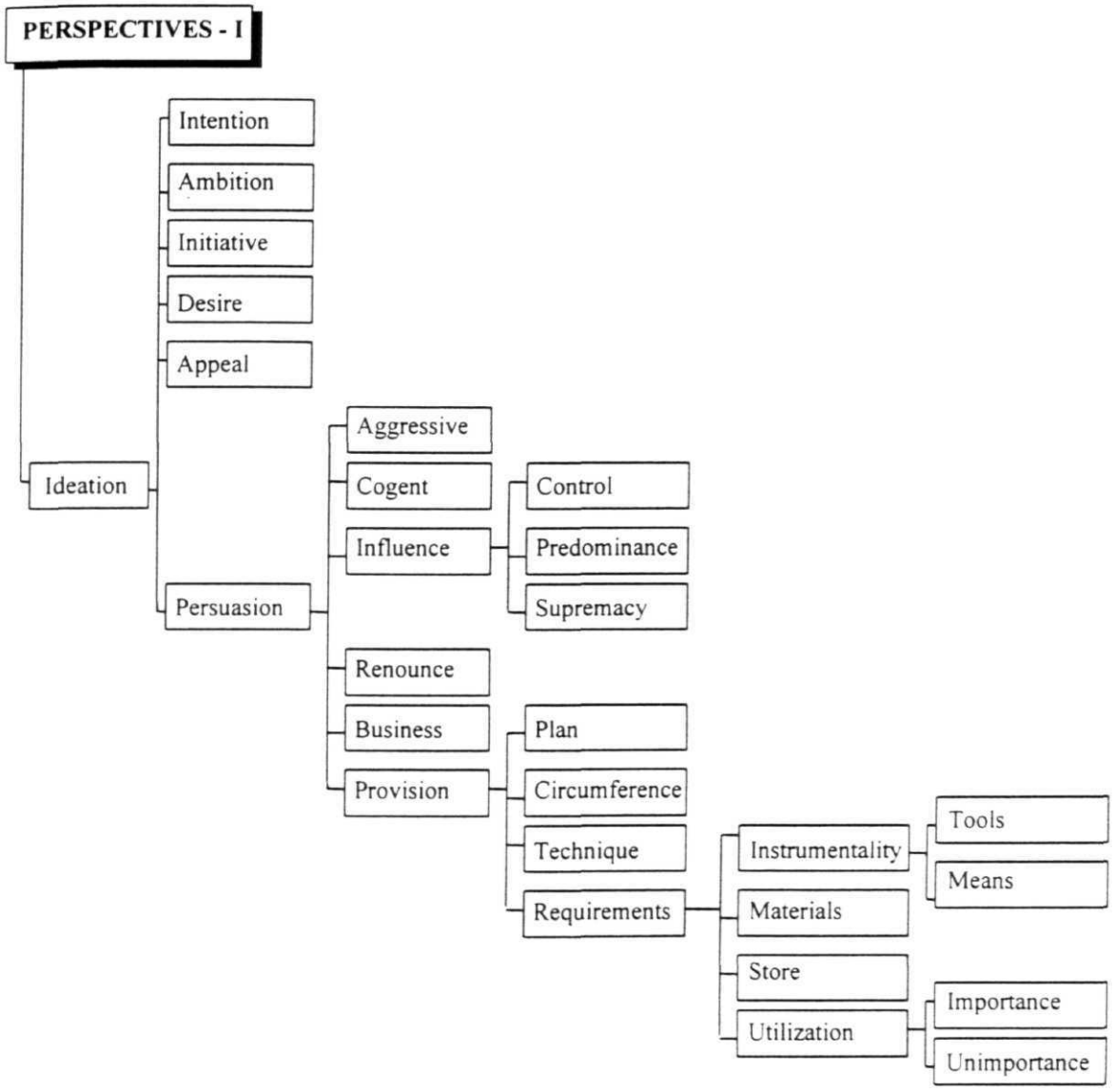
4. DOMAIN : ACTION + VOLITION,MOVEMENT,***POSITION***



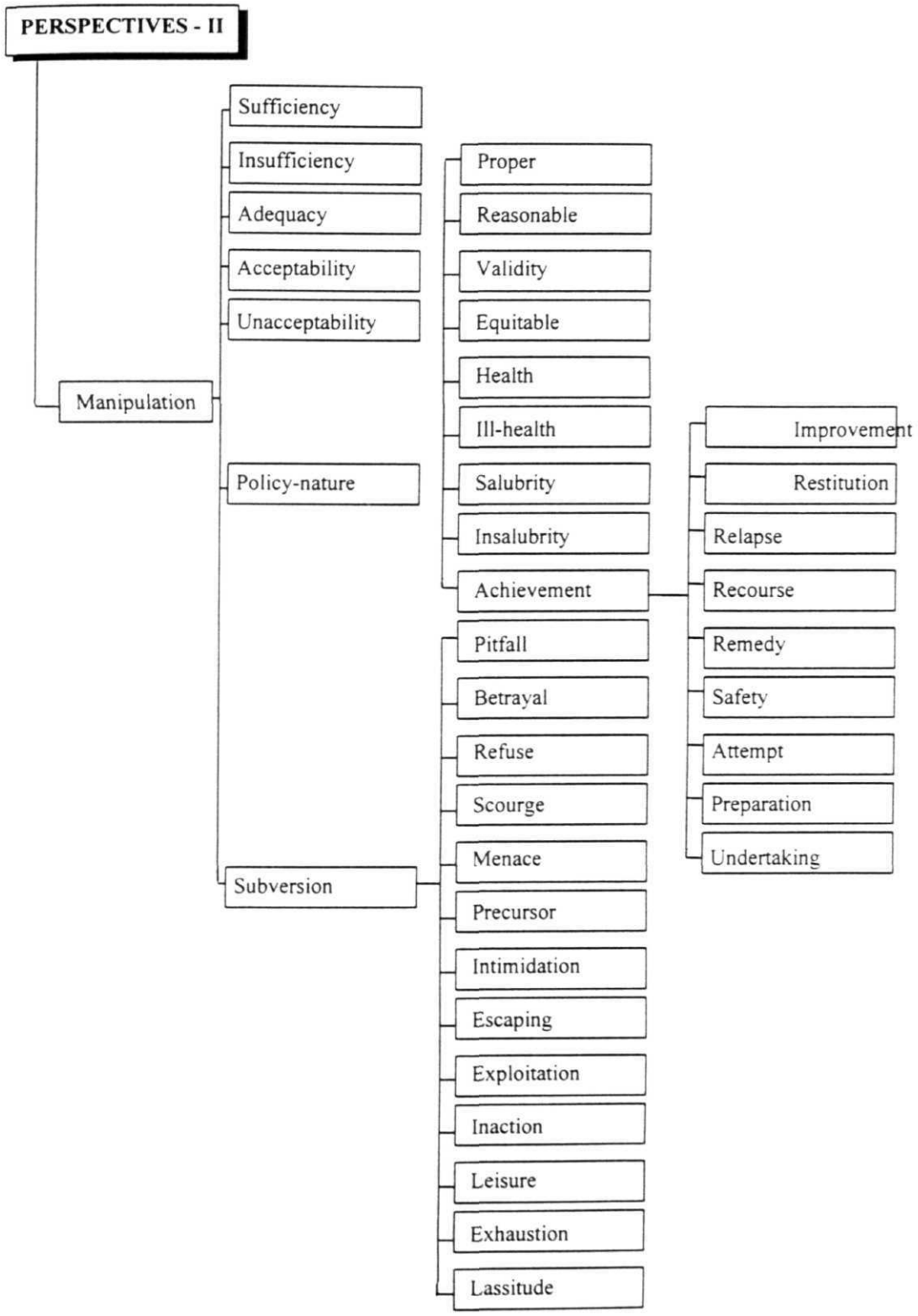
4. DOMAIN : ACTION + VOLITION,VOLITION,***PSYCHIC ACTS***



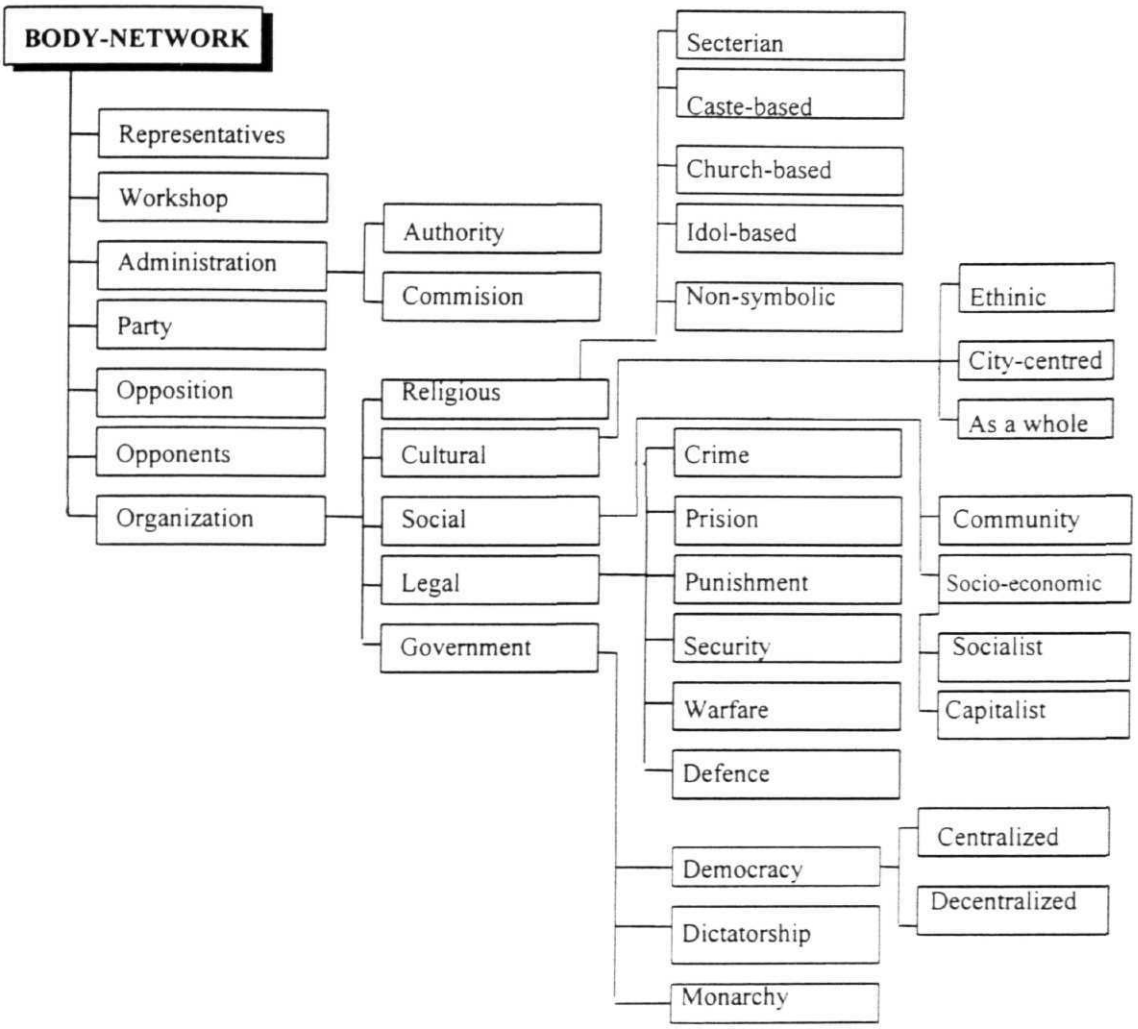
4. DOMAIN : ACTION + VOLITION,VOLITION,***PERSPECTIVES* -I**



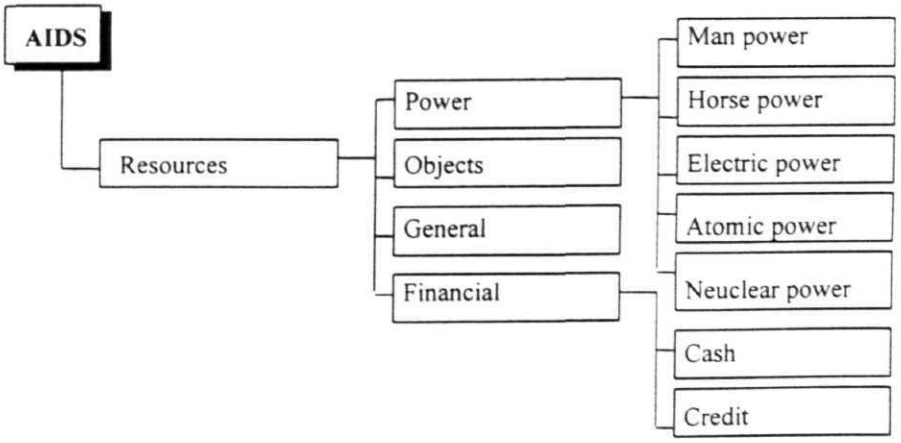
4. DOMAIN : ACTION + VOLITION,VOLITION,***PERSPECTIVES*** **-II**



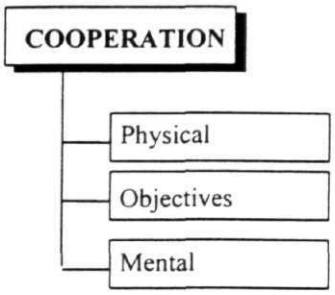
4. DOMAIN : ACTION + VOLITION,VOLITION,***BODY-NETWORK***



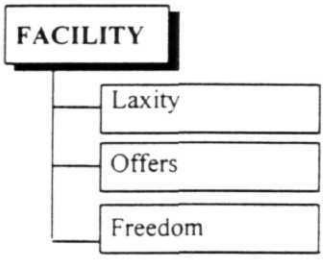
4. DOMAIN : ACTION + VOLITION,FACILIATION,***AIDS***



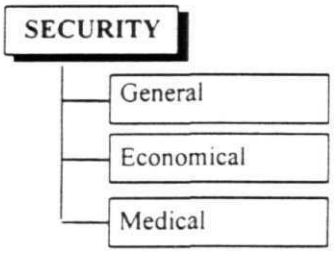
4. DOMAIN : ACTION + VOLITION,FACILIATION,***COOPERATION***



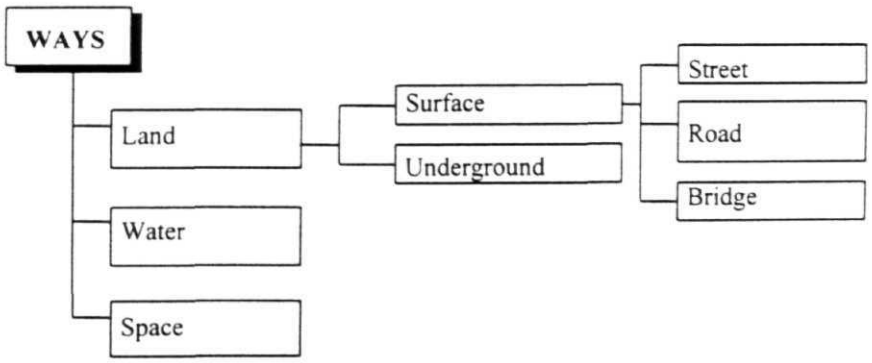
4. DOMAIN : ACTION + VOLITION,FACILIATION,***FACILITY***



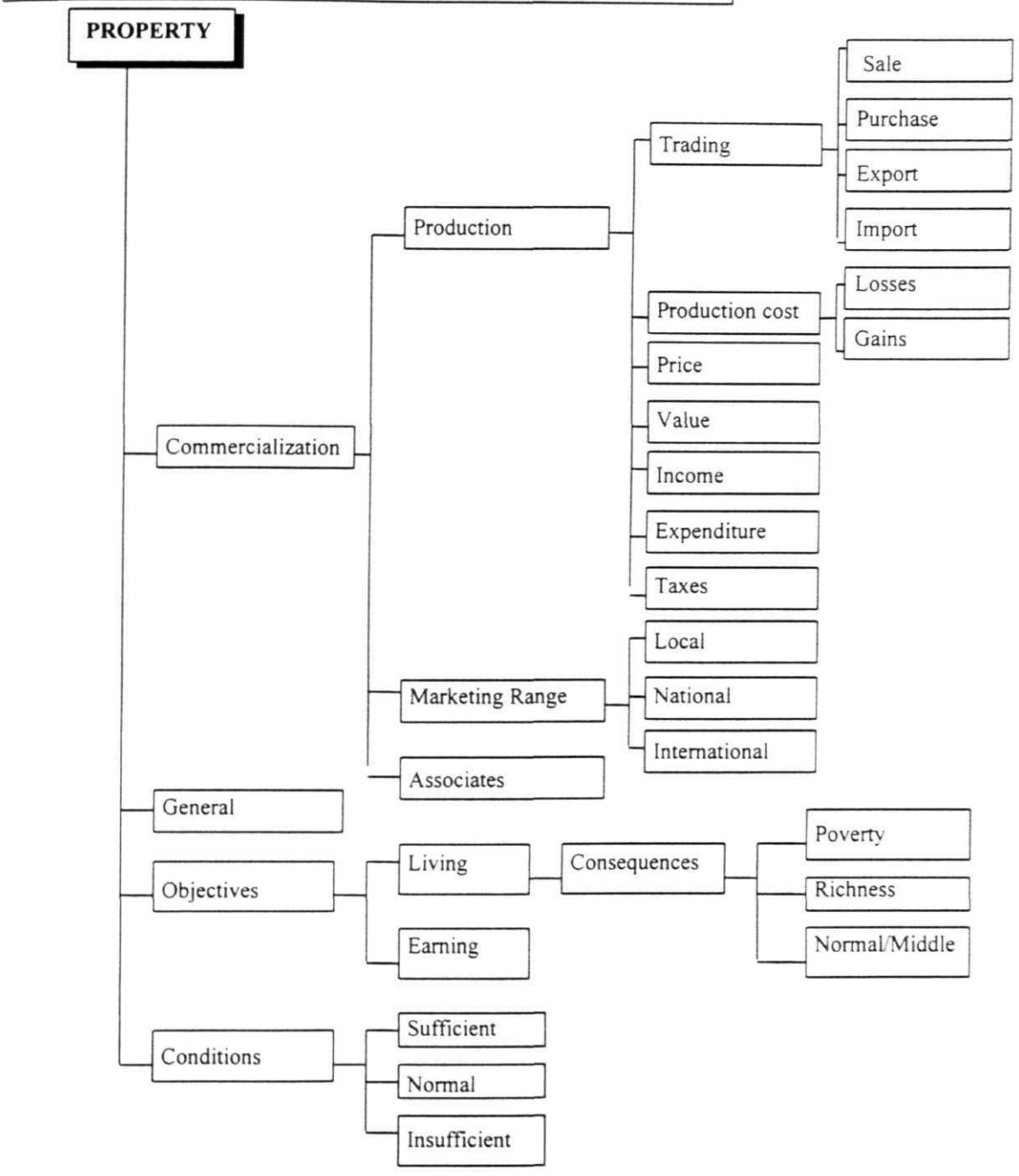
4. DOMAIN : ACTION + VOLITION,FACILIATION,***SECURITY***



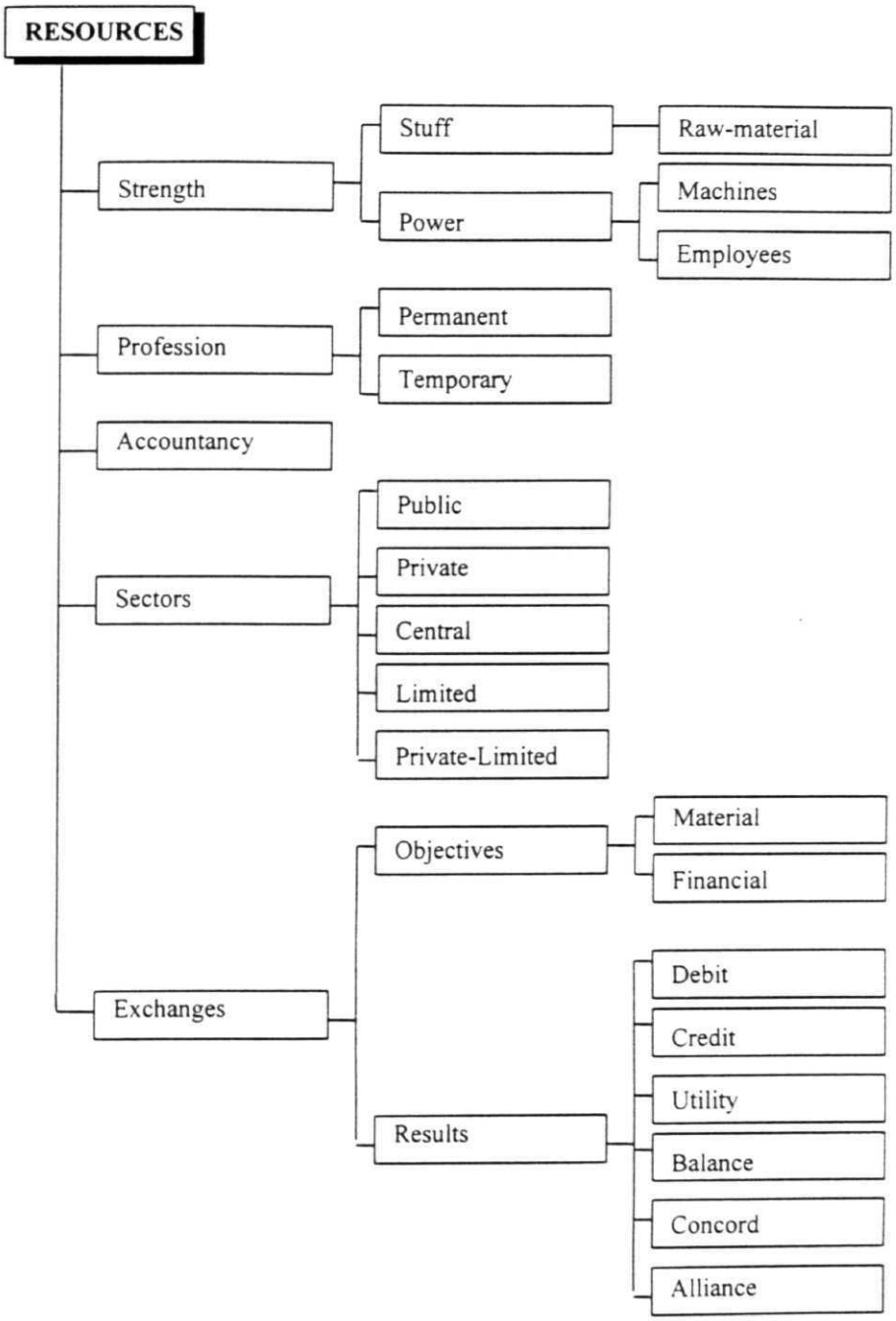
4. DOMAIN : ACTION + VOLITION,FACILIATION,***WAYS***



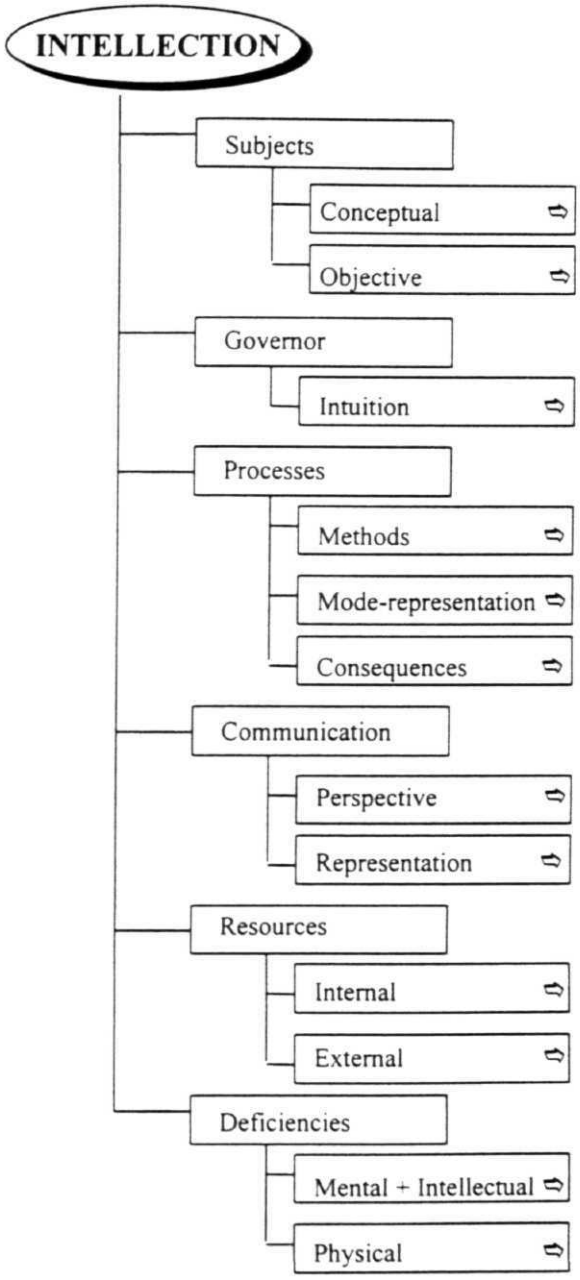
4. DOMAIN : ACTION + VOLITION,POSSESSION,***PROPERTY***



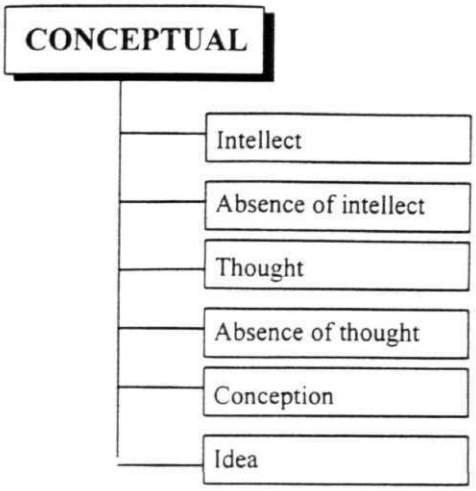
4. DOMAIN : ACTION + VOLITION,POSSESSION,***RESOURCES***



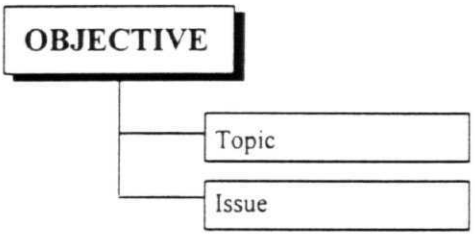
3.2.5. DOMAIN : *INTELLECTION*



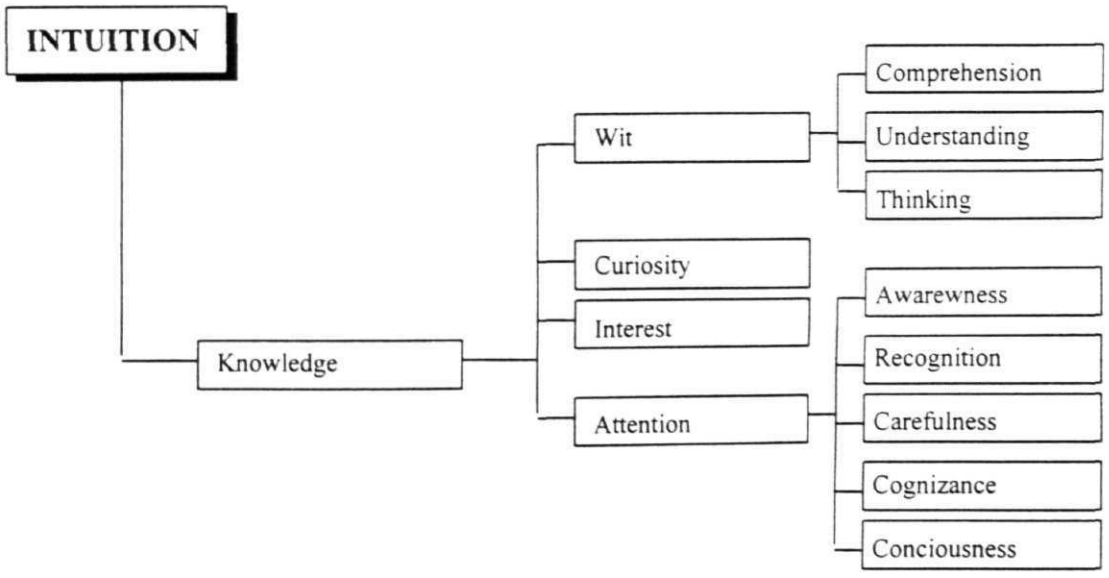
5. DOMAIN : INTELLECTION,SUBJECTS,***CONCEPTUAL***



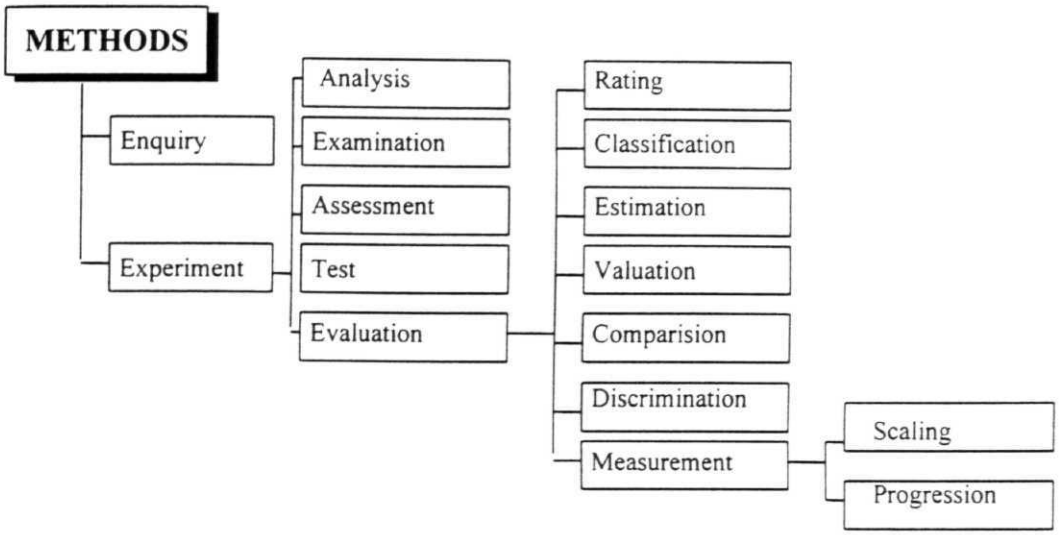
5. DOMAIN : INTELLECTION,SUBJECTS,***OBJECTIVE***



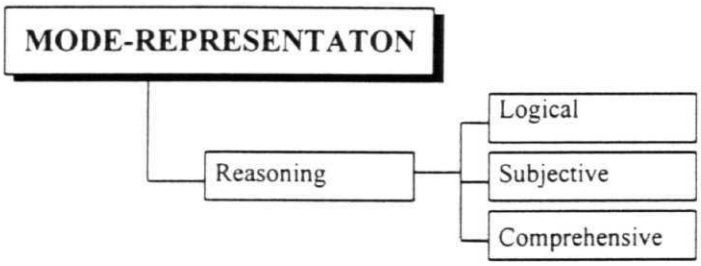
5. DOMAIN : INTELLECTION,GOVERNORS,***INTUITION***



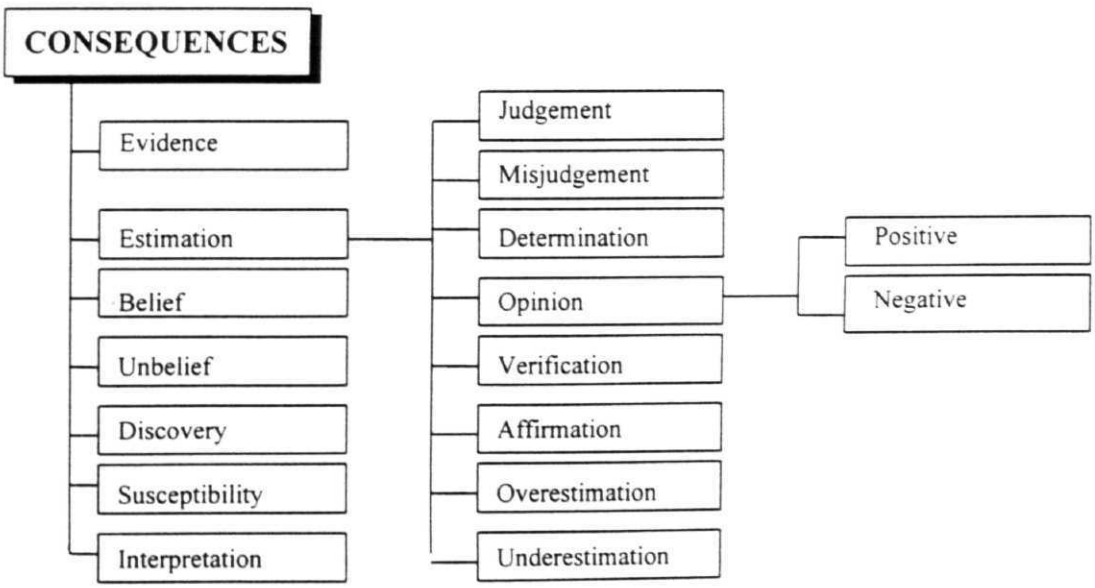
5. DOMAIN : INTELLECTION,PROCESSES,***METHODS***



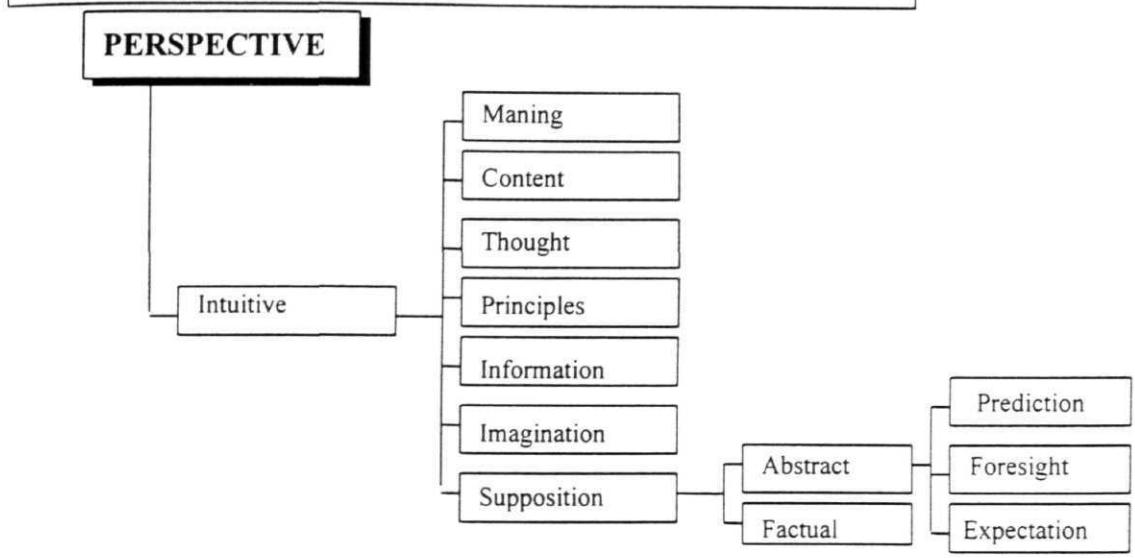
5. DOMAIN : INTELLECTION,PROCESSES,***MODE-REPRESENTATION***



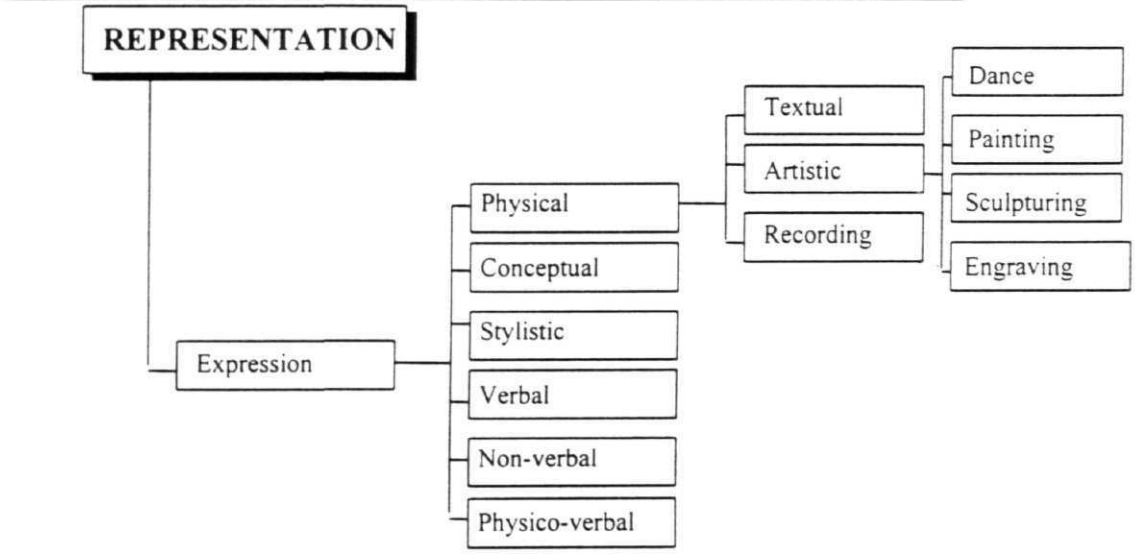
5. DOMAIN : INTELLECTION,PROCESSES,***CONSEQUENCES***



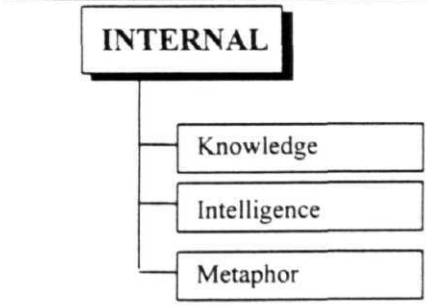
5. DOMAIN : INTELLECTION,COMMUNICATION,***PERSPECTIVE***



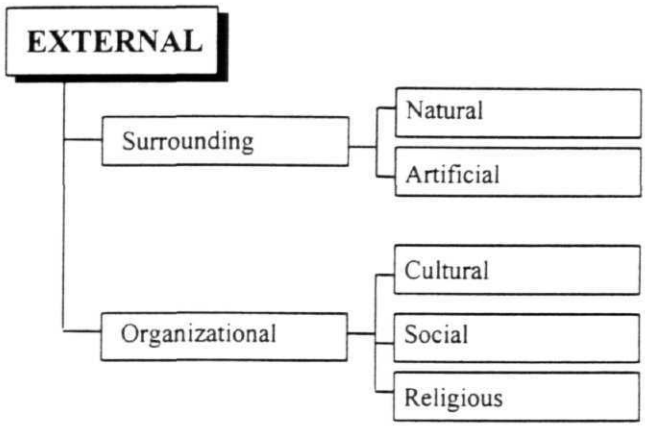
5. DOMAIN : INTELLECTION,COMMUNICATION,***REPRESENTATION***



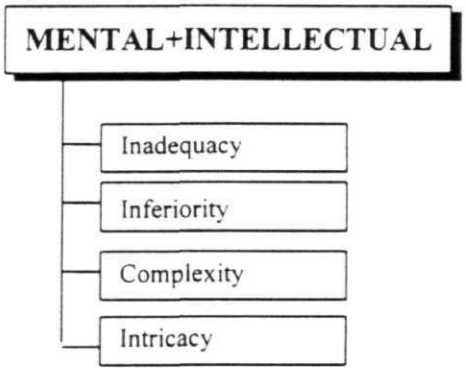
5. DOMAIN : INTELLECTION,RESOURCES,***INTERNAL***



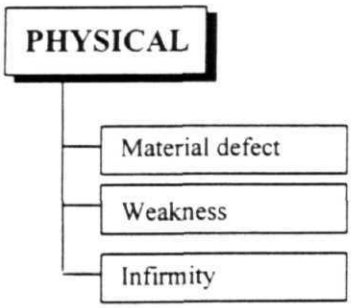
5. DOMAIN : INTELLECTION,RESOURCES,***EXTERNAL***



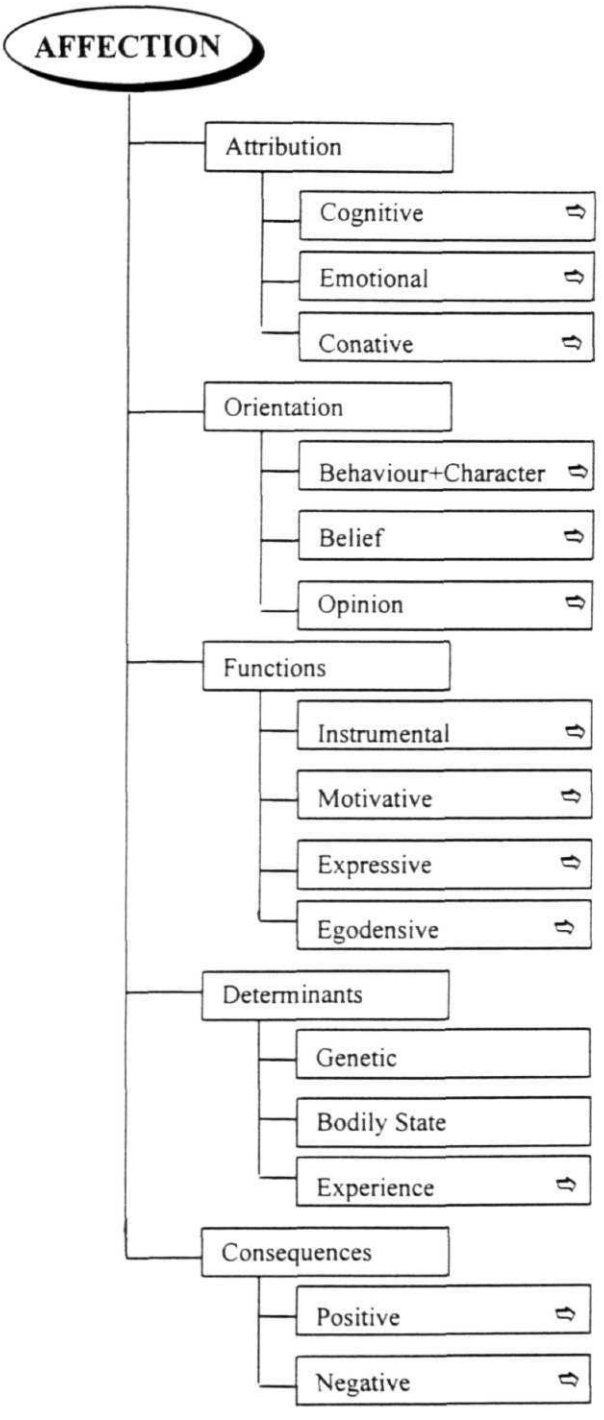
5. DOMAIN : INTELLECTION,DEFICIENCIES,***MENTAL+INTELLECTUAL***



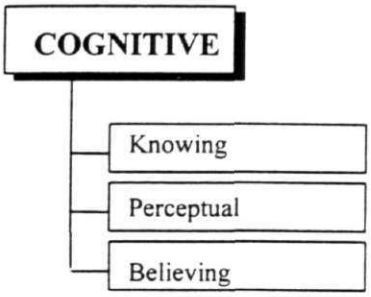
5. DOMAIN : INTELLECTION,DEFICIENCIES,***PHYSICAL***



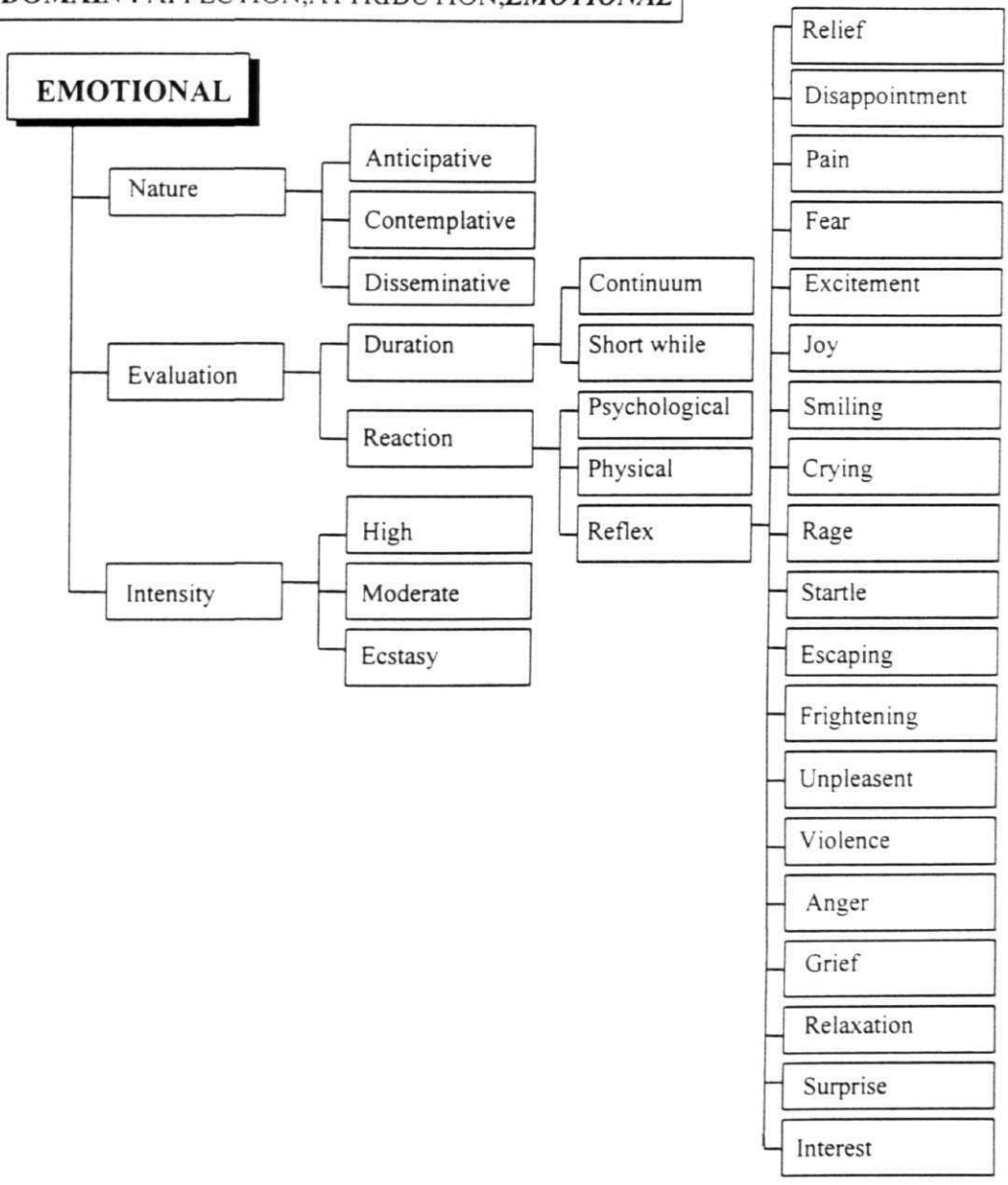
3.2.6. DOMAIN : *AFFECTION*



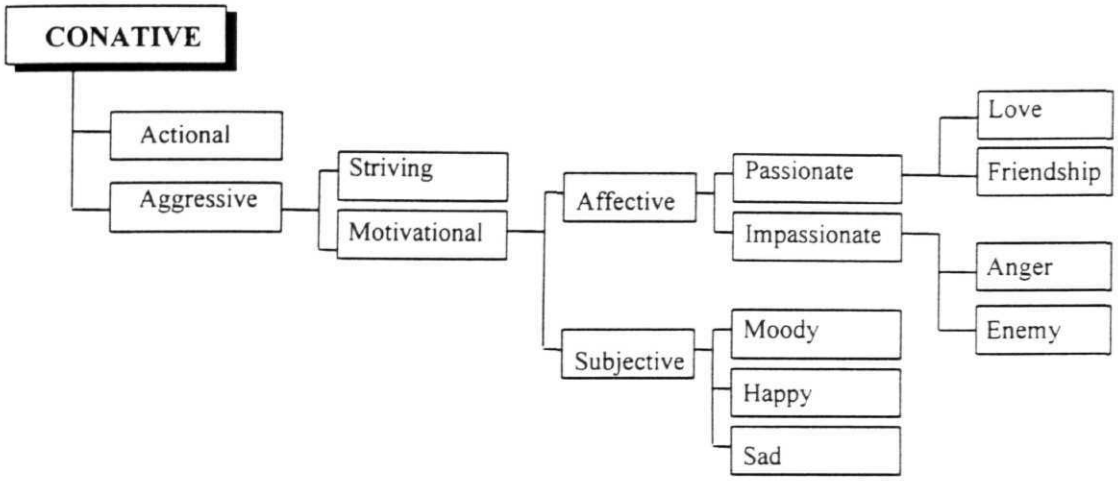
6. DOMAIN : AFFECTION,ATTRIBUTION,***COGNITIVE***



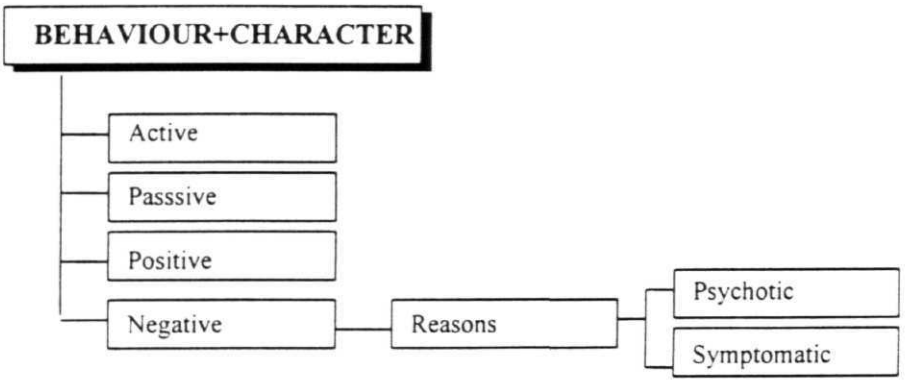
6. DOMAIN : AFFECTION,ATTRIBUTION,***EMOTIONAL***



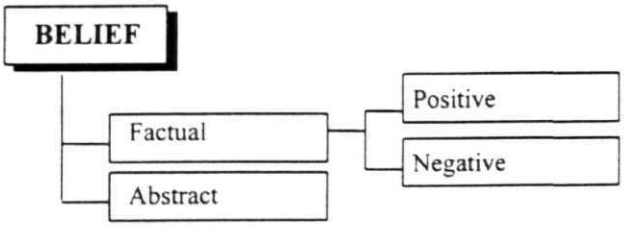
6. DOMAIN : AFFECTION,ATTRIBUTION,***CONATIVE***

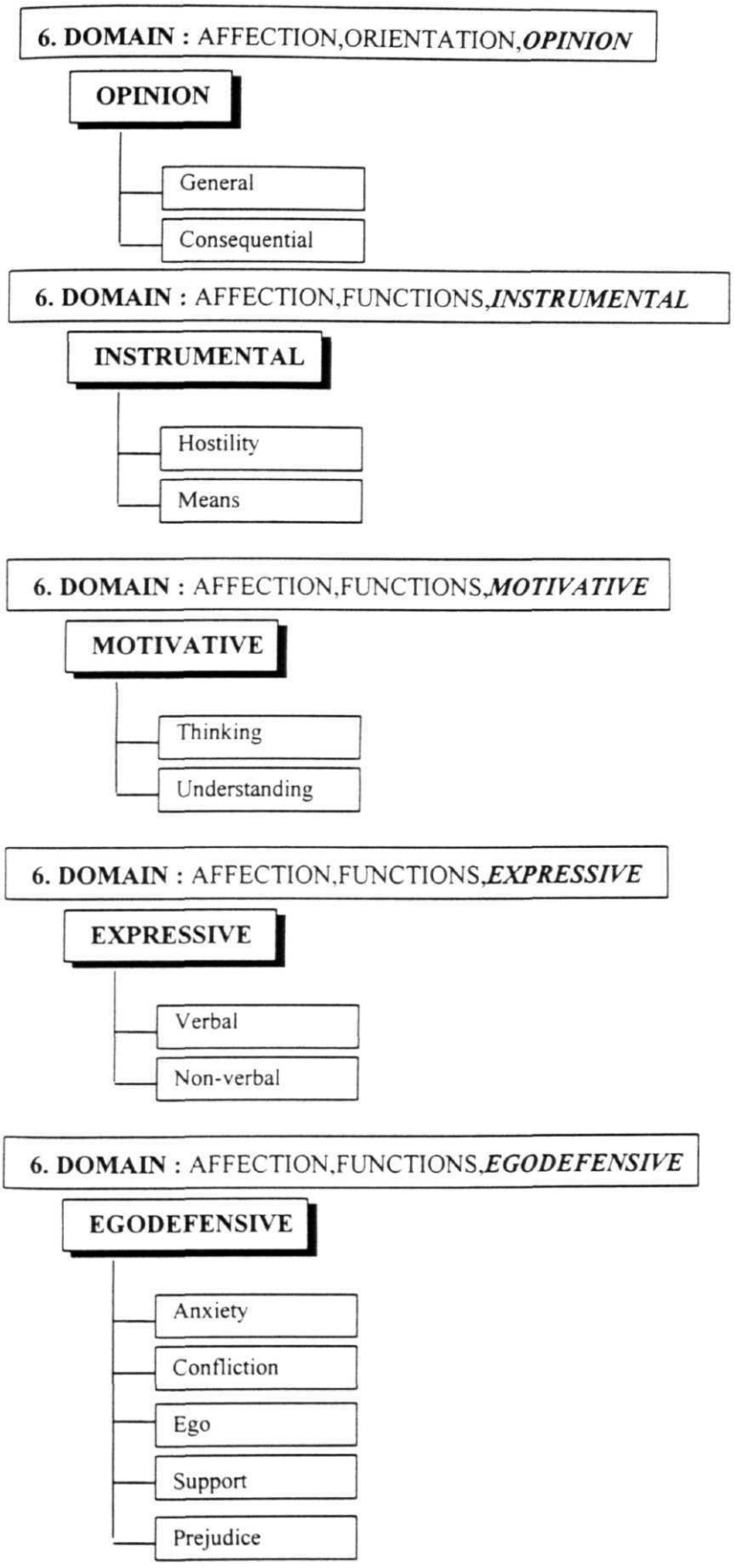


6. DOMAIN : AFFECTION,ORIENTATION,***BEHAVIOUR+CHARACTER***

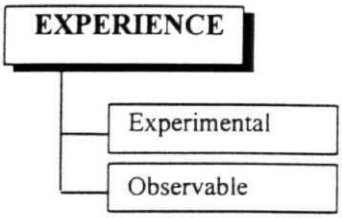


6. DOMAIN : AFFECTION,ORIENTATION,***BELIEF***

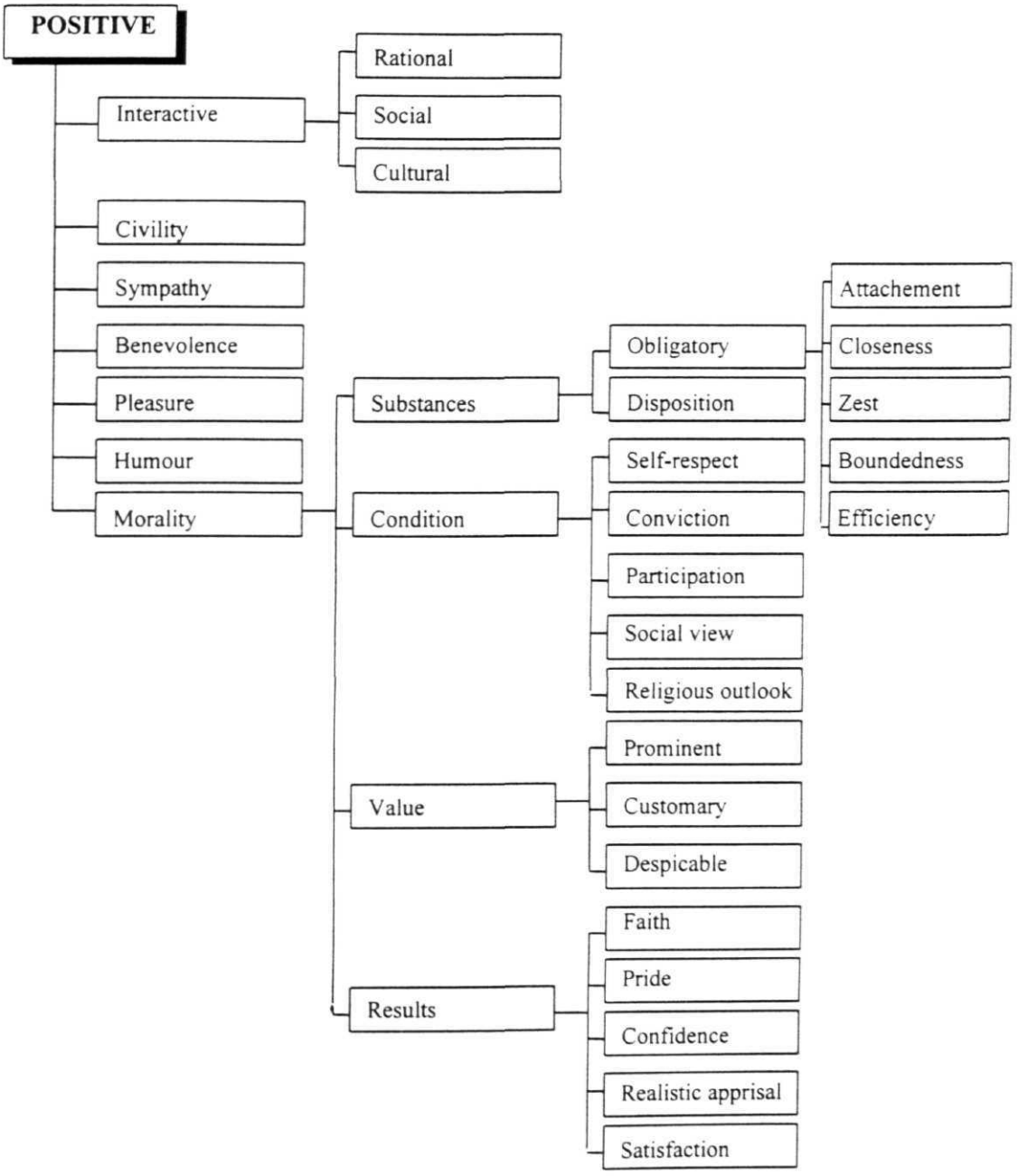



6. DOMAIN : AFFECTION,ORIENTATION,OPINION
OPINION
General
Consequential
6. DOMAIN : AFFECTION,FUNCTIONS,INSTRUMENTAL
INSTRUMENTAL
Hostility
Means
6. DOMAIN : AFFECTION,FUNCTIONS,MOTIVATIVE
MOTIVATIVE
Thinking
Understanding
6. DOMAIN : AFFECTION,FUNCTIONS,EXPRESSIVE
EXPRESSIVE
Verbal
Non-verbal
6. DOMAIN : AFFECTION,FUNCTIONS,EGODEFENSIVE
EGODEFENSIVE
Anxiety
Confliction
Ego
Support
Prejudice

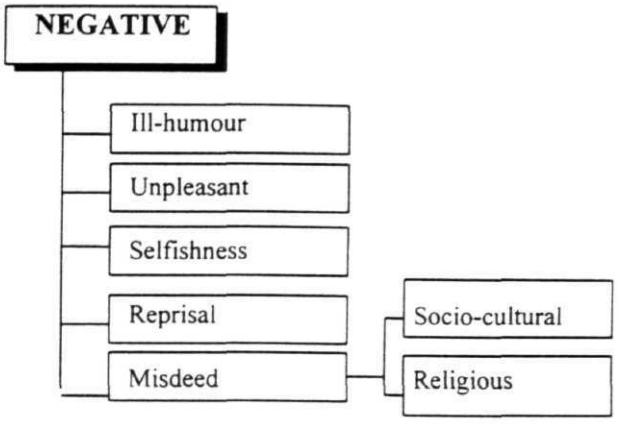
6. DOMAIN : AFFECTION,DETERMINANTS,***EXPERIENCE***



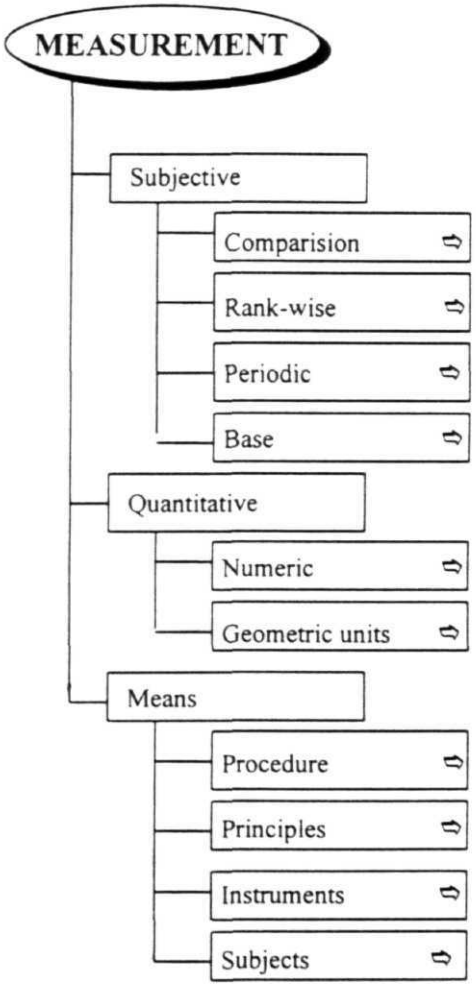
6. DOMAIN : AFFECTION,CONSEQUENCES,***POSITIVE***



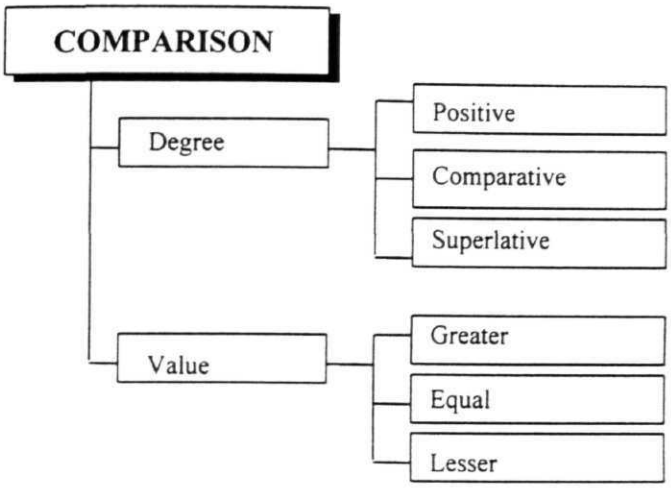
6. DOMAIN : AFFECTION,CONSEQUENCES,***NEGATIVE***



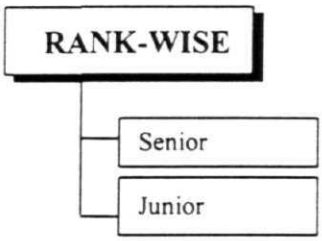
3.2. 7. DOMAIN : *MEASUREMENT*



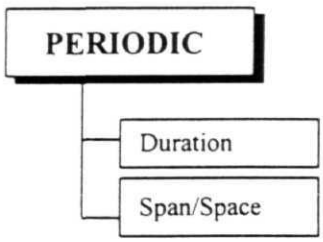
7. DOMAIN : MEASUREMENT,SUBJECTIVE, ***COMPARISON***



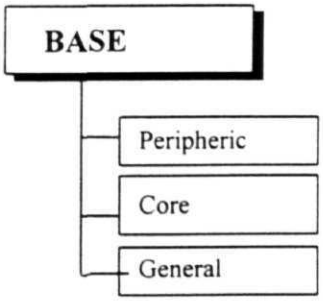
7. DOMAIN : MEASUREMENT,SUBJECTIVE,***RANK-WISE***



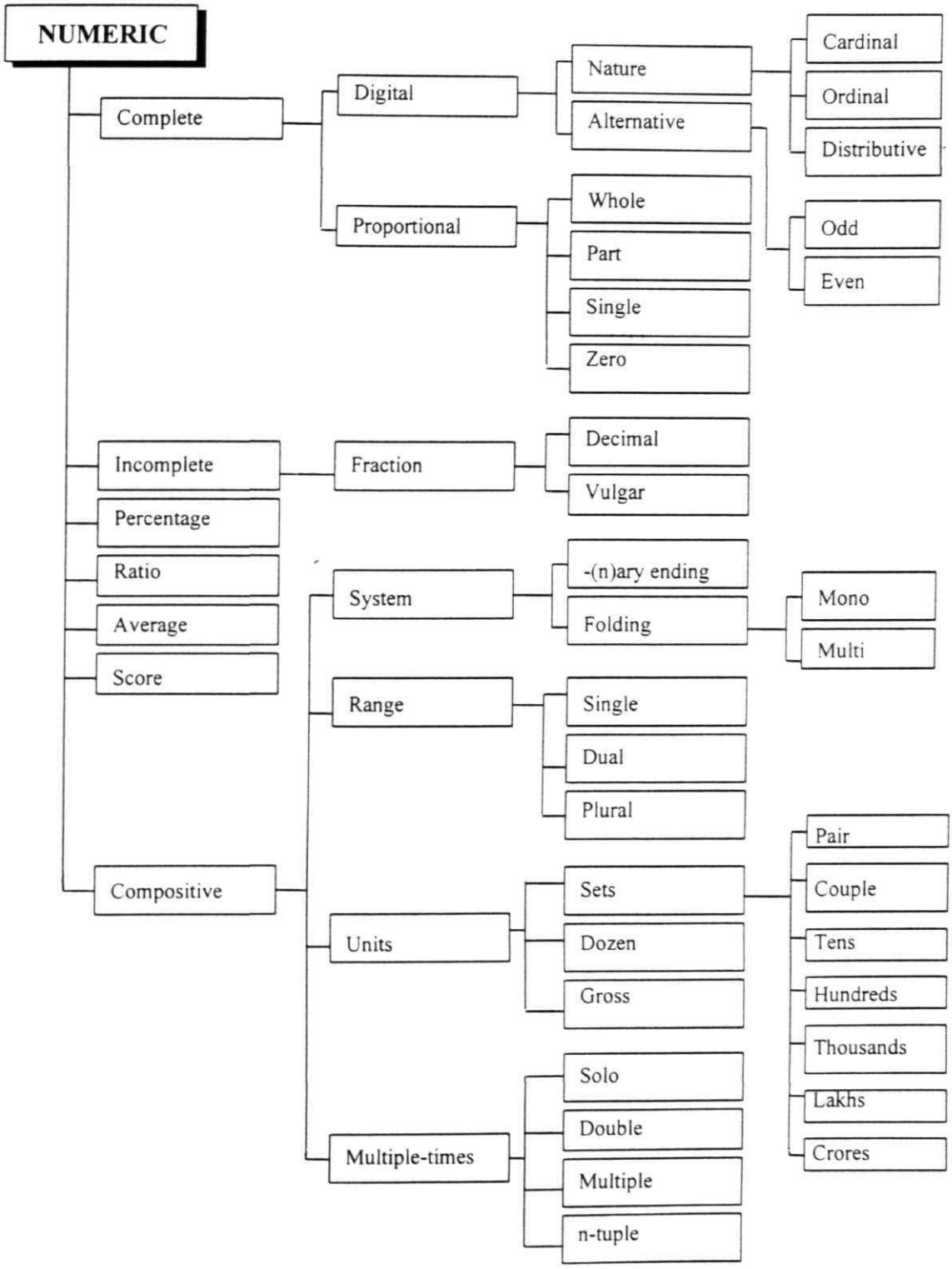
7. DOMAIN : MEASUREMENT,SUBJECTIVE,***PERIODIC***



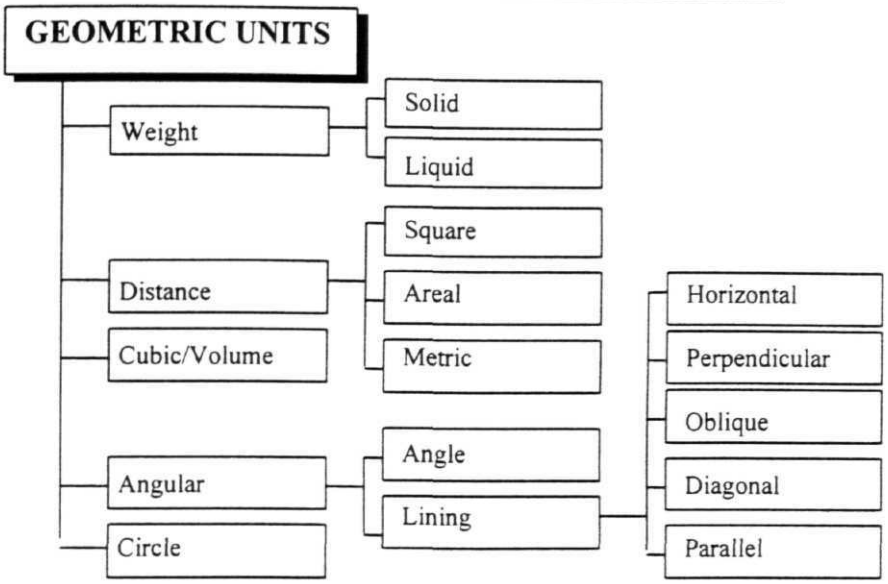
7. DOMAIN : MEASUREMENT,SUBJECTIVE,***BASE***



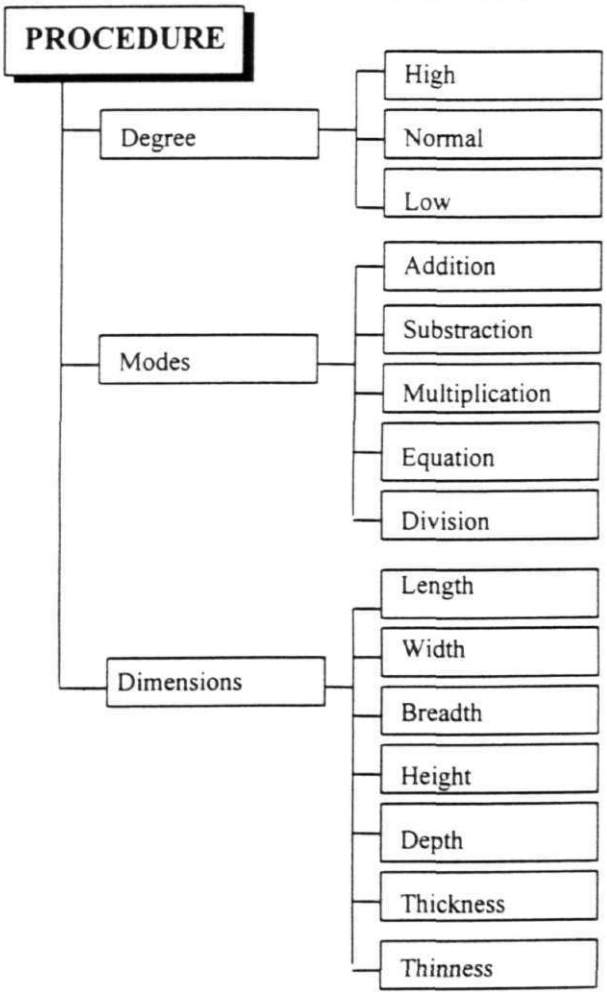
7. DOMAIN : MEASUREMENT,QUANTITATIVE,***NUMERIC***



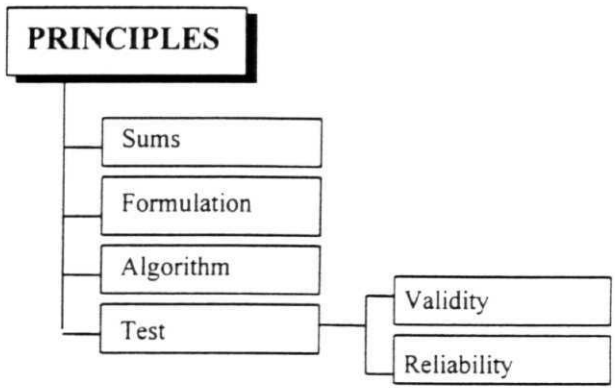
7. DOMAIN : MEASUREMENT,QUANTITATIVE,***GEOMETRIC UNITS***



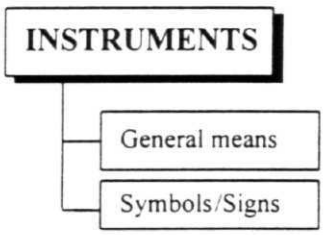
7. DOMAIN : MEASUREMENT,MEANS,***PROCEDURE***



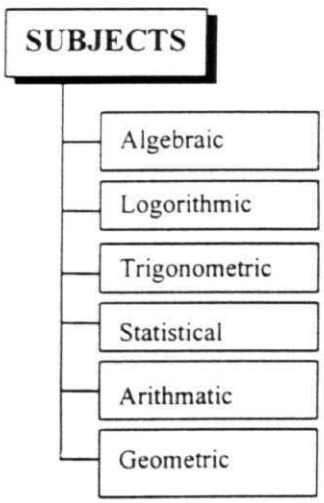
7. DOMAIN : MEASUREMENT,MEANS,***PRINCIPLES***



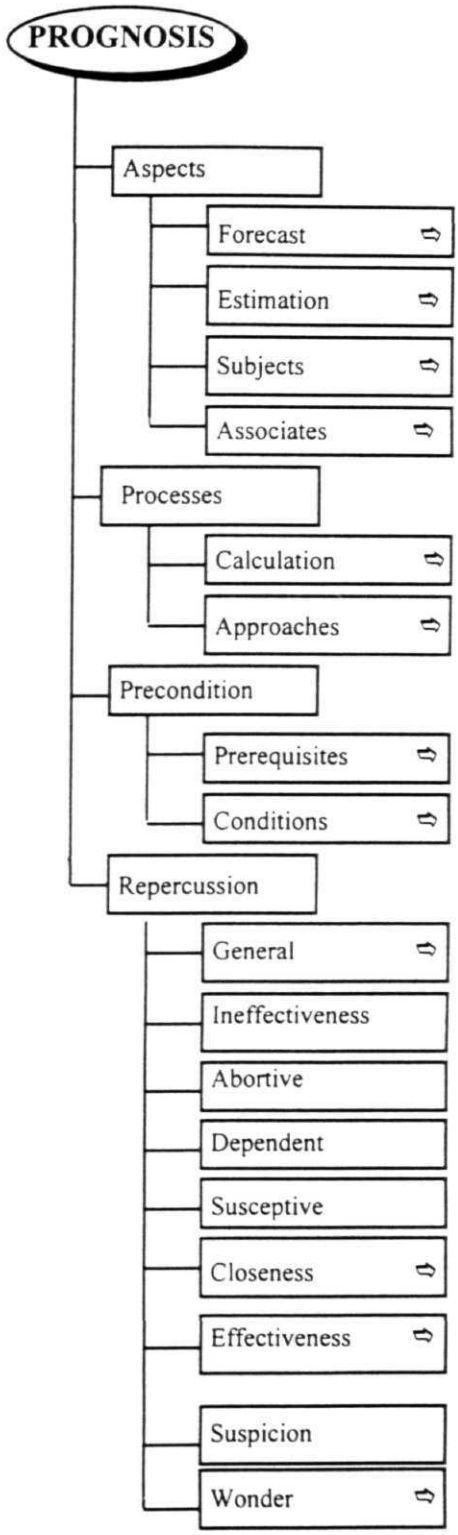
7. DOMAIN : MEASUREMENT,MEANS,***INSTRUMENTS***



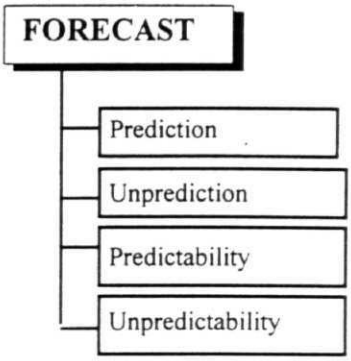
7. DOMAIN : MEASUREMENT,MEANS,***SUBJECTS***



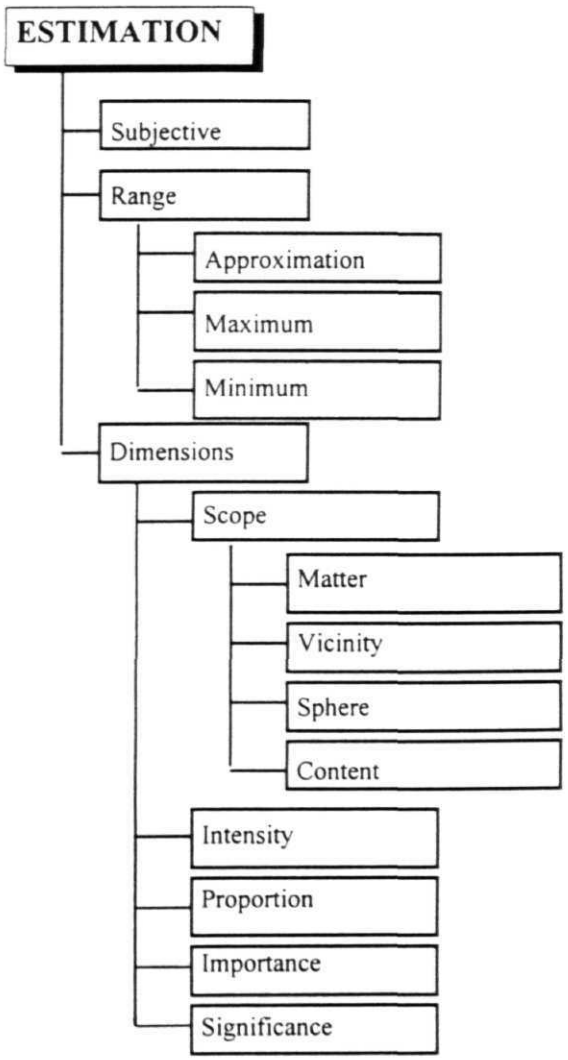
3.2.8. DOMAIN : *PROGNOSIS*



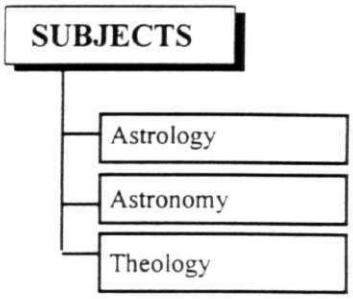
8. DOMAIN : PROGNOSIS,ASPECTS,***FORECAST***



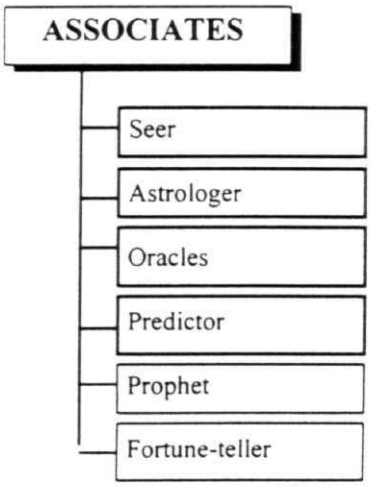
8. DOMAIN : PROGNOSIS,ASPECTS,***ESTIMATION***



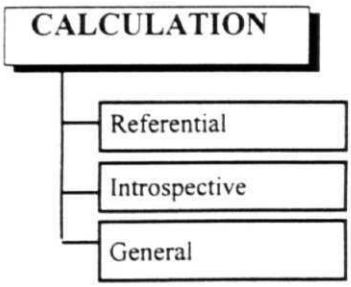
8. DOMAIN : PROGNOSIS,ASPECTS,***SUBJECTS***



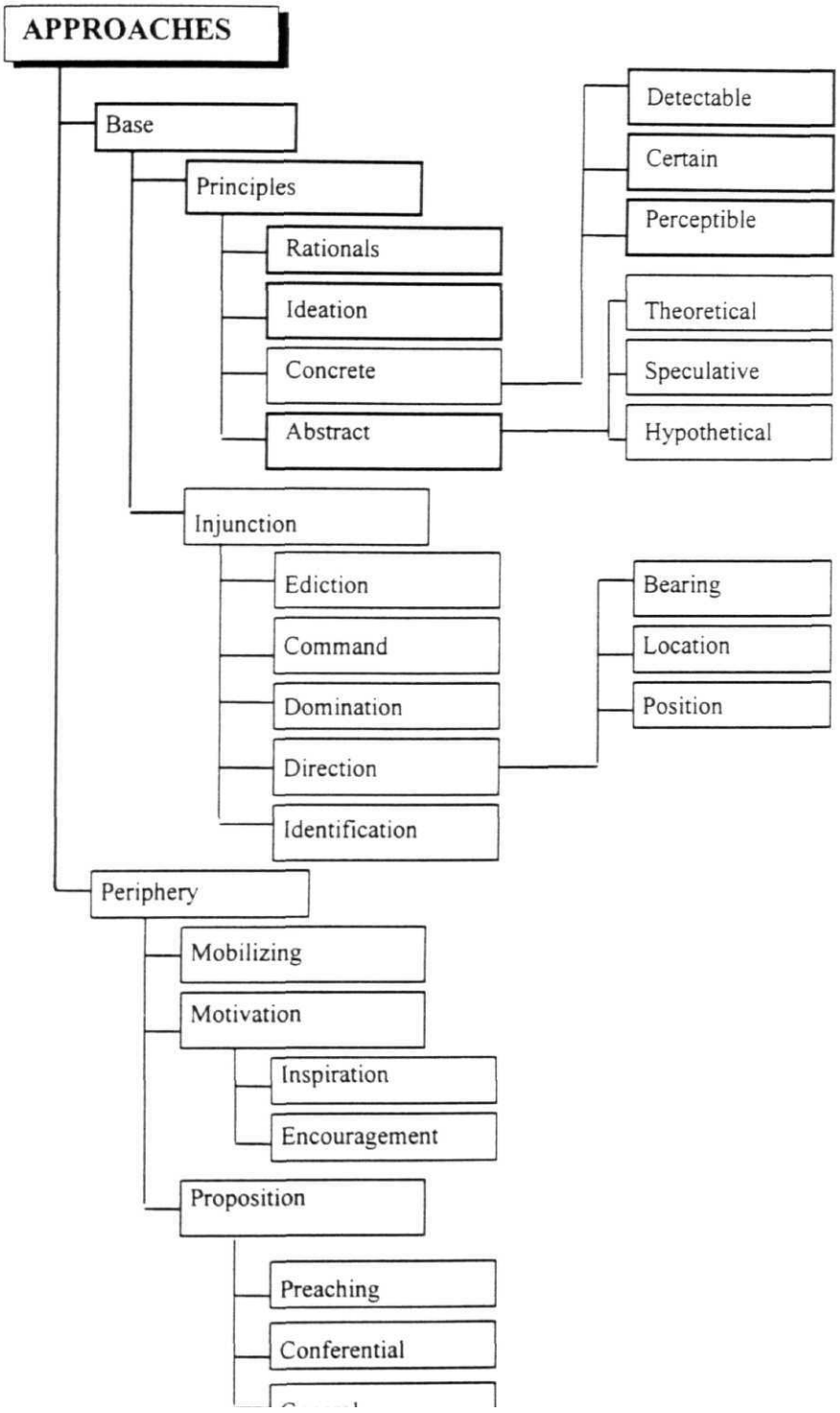
8. DOMAIN : PROGNOSIS,ASPECTS,***ASSOCIATES***



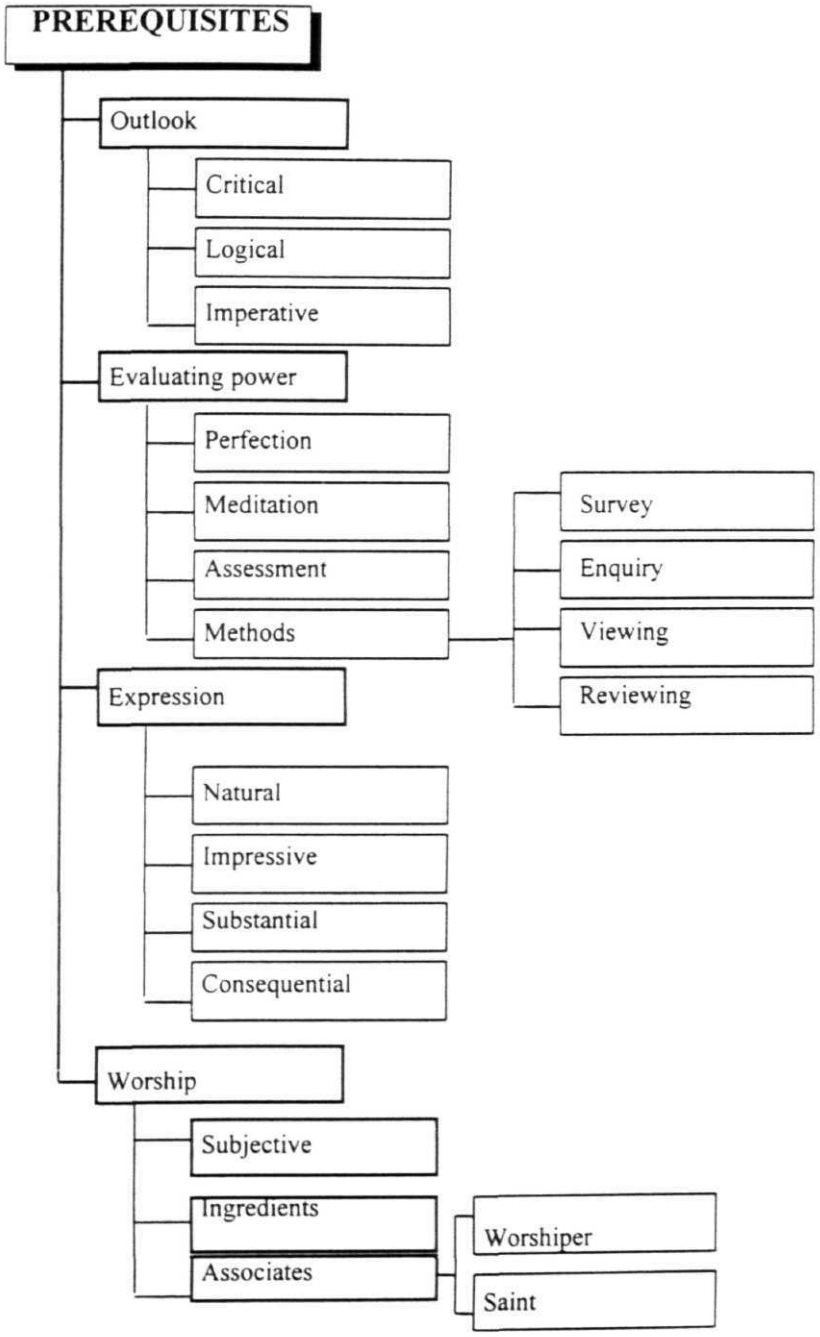
8. DOMAIN : PROGNOSIS,PROCESSES,***CALCULATION***



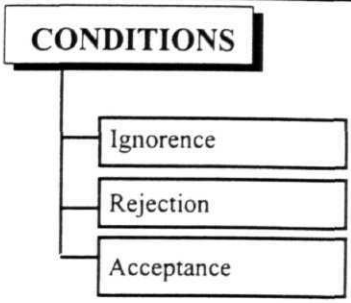
8. DOMAIN : PROGNOSIS,PROCESSES,***APPROACHES***



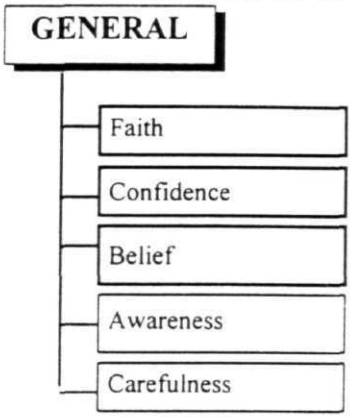
8. DOMAIN : PROGNOSIS,PRECONDITIONS,***PREREQUISITES***



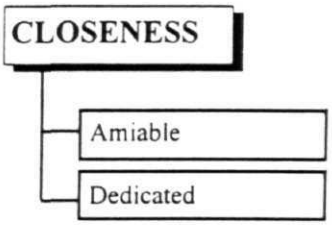
8. DOMAIN : PROGNOSIS,PRECONDITION,***CONDITIONS***



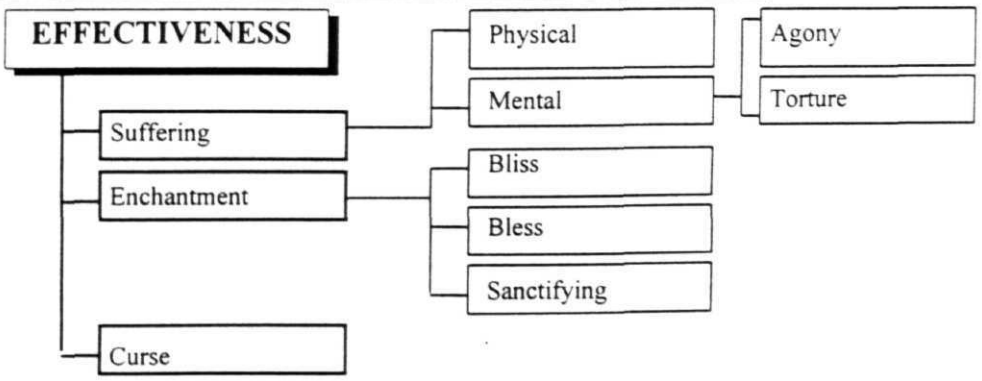
8. DOMAIN : PROGNOSIS,REPERCUSSION,***GENERAL***



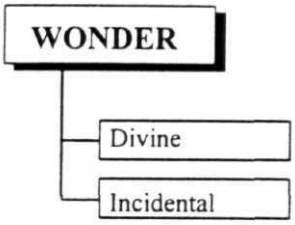
8. DOMAIN : PROGNOSIS,REPERCUSSION,***CLOSENESS***



8. DOMAIN : PROGNOSIS,REPERCUSSION,***EFFECTIVENESS***



8. DOMAIN : PROGNOSIS,REPERCUSSION,^0M)£tf



33. GENERAL CHARACTERISTICS

Certain general characteristics emerge out of these detailed categorization and classification, and they are as follows :

1. These features are highly specific (specific to South Asian context) category labels (such as 'Joint Family' or "Taluka") in this design which need to be modified if the broad structure is to be adopted for other language-groups.

2. The same category-labels may appear under different domains and categories - and this cross-referencing power (or we may call it "discursiveness") is what makes a thesaurus more usable.

3. Our time does have reflection on our design today. If one looks at Roget's design, the progress of natural sciences upto his time (over a century ago) is clearly reflected in his category-1 a be Is such as 'matter', etc. Similarly. 'World war' or 'Biological changes" or **'Star war'** as given in SELF classes cannot be avoided today.

3.4. APPLICATION : SEMANTIC GRAVITY

Thesaurus is concerned with the formal and semantic, as they are interrelated, refemce of words in terms of accurate use of the words in a given context in linguistically and lexicographically acceptable fashion. The aim of a thesaurus is to mark out the precise and essential information about the forms and their semantic significations based on the suitability of the contexts. The sense assignement of the lexical item possibily extended through the semblance *of* functions which may vary from one sphere to another sphere. If meaning of a lexical item is changed according to the various contexts, the semantically classified thesaurus extends separate semantic entities for a given word. Semantic classification may be an excellent methodological device to apply to the entry substance-representation. The degree of complexity lies in the account of the very nature. If it occurs once and twice, the complexity will be very low. But in contrast, a word occurs in many contexts with the same nunber of semantic assignment, the degree of complexity will be respectively high. The semantic shadow is a substantiality of the interpretation which is logically related to the context in wich the word occurs. Such conspicuous semantic shadow must be taken into account by defining the usage and proper semantic distinction .

A thesaurus based on the semantic classification enchants with vocabularies of a language from an inflectional aspect covering the total sum records (i.e. lexical items) of the language along with the changes have taken place right from the earliest period to the present. It takes into account their formal and semantic aspects and their time frame without giving any overt information. Every lexicon has its own model of entry formation, which is clearly evident especially in the thesaurus-building which has clearly two models - alphabetic and semantic (as discussed previously), depending upon the nature of arrangement of lexical entries. It is, obviously, the physical part of the lexicon dealing with the visible drafting shape resulting as an output and the non-physical part undertaking the whole precess put into practice as an input issue. The whole vocation of preperation of semantic thesaurus undergoes several sets of assembling activities, the assemblage of attributes, undersigned processes, drafting devices, and exemplary approaches which are viewed in elucidation. The creation of a semantically classified thesaurus comprises some hierarchical categories. Just for example, subject, sense, sub-sense, and lexical items used as synonyms of headword.

The lexical entries, being included in semantically classified theasuarus. share some general features which are identified subjectwise by entitling their respective subject. Therefore, it consists of merely the major sense of the word from which it belongs to, as for example, literature, technology, science, and commerce. Headword is a lexicographic technical term defined as 'a lexical unit/entry' which is considered as a free form. Otherwise, it is called a "lemma". It is arranged in 'canonical' or "ideal' form. Indeed, the head-entry always comes at initial position as free lexical unit. Moreover, the derivative forms which normally come under the respective headword (head entry) as sub-headword or sub-entry are considered as free forms in this research and are placed in the head-entry position.

Words, in fact, belong to a particular group having uniqueness in semantico-pragmatic application and they share a number of common features. Due to this sharing of the same or similar semantico-pragmatic assigned features, they come under a particular sense category. For example, vvavahAra, AcaraNa, gatimAna, samay. vastu, jAnavara, and pakSI are the distinguished senses of the Hindi word 'can*cala\ The sense-assignment can be further classified into sub-sense-assignments having uniqueness in semantico-pragmatic application, for instance, the cAlaka functions as sub-sense of the sense AcaraNa of the headword "can*cala. There is. indeed, hierarchical relationship between each higher category (functioning as hyperonym or superordinate) and just lower categories (functioning as hyponyms or ordination). That is why, sometimes the sense can function as a sense-cume sub-sense. The occurence of sense, sub-sense and lexis can take place at any stage with different functions within the whole thesaurus-body. For instance, a word in lexical sphere in particular context can also occur either in sense, or sub-sense, or lexical sphere in another context(s) with different functions.

3.5. SUMMARIZING NOTES

The semantically classified thesaurs as a comprehensive insightful reference book requires that the maximum help is provided to the users in comprehending the detailed semantic shades of each word. Thesaurus takes care of multidynamic implication *of* semantic representation of all lexical units. The format proposed here is based on the clssification done according to the semantic features in order to overcome the limitation as stated by Laird (1990 : ix), "... any one who uses a thesaurus should have at least one other wordbook to tell hime more about the words he has found in the thesaurus that are likely condidates." The model presented here being both word-finder and synonym-locator, this criticism is possible to overcome here.

CHAPTER 4

ON-LINE NETWORK

...yet until very recently dictionaries for lexicons as linguists usually call them) for Natural Language Processing systems have by and large been poor sister of computational linguistic ***research"***

-Boguraev (1989:1).

4.0. PRELIMINARIES : *COMPUTATIONAL COMPILATION*

The computer, in terms of compilation, performs its tasks automatically as "post-processor' as well as 'pre-processor'. As a post-processor, it performs the tasks of programme-implementation whereas a pre-processor, it is used for data-storage. As Boguraev (1991:163) states, "Computers are also extensively used both before and after a new dictionary has taken shape. Potentially mundane, but nonetheless critically required processes as maintaining control of e.g. style, consistency, and uniformity of notion and convention can be carried out by a programme running autonomously as a post-processor. On the other hand, and specially where wholly new dictionaries are concerned, management and pre-processing of large corpora of raw text data are an essential prerequisite for dictionary making projects." This chapter consists of three points : a brief introduction about the Visual Basic 5.0 version , GUI : window-displays, and database used for the Hindi thesaurus on display.

4.1. **VISUAL** BASIC : *ENVIRONMENT TO APPLICATION*

The Graphical User Interface (GUI) being developed for the present thesaurus module has been designed on the visual platform with the help of the Visual Basic 5.0. The projecting view of visual effect is, therefore, as insightful as the word-representation. The system requirements, as per the minimum cardinal setup, are as follows :

IBM -computer with 80486 processor or higher capable processor, but Pentium will be easier in accesibility.
Windows version, viz. Windows' 95 or later version.
8 MB of RAM.
Hard disk reqirement 90 MB-170 MB for VB installation.
30 MB free space in hard disk is required to run VB verson 5.0.
Other assembled requirements are :
A mouse
CD_ROM drive.
Colour Monitor - EGA, VGA, or other compatible display system

If the computer is equipped with the above mentioned system-prerequisites, the accessibility of VB will be faciliated in its application. It is easier to design the screen-windows in VB. As per its readymade amenities is concerned, programming in VB is much more moderate and comfortable which can be carried out without having much computational programming expertise. Though VB has multi-dimensional facilities in its application of desigining and developing the software packages, the restricted information about the application to the thesaurus will be accorded. The features of VB as a desigining and developing package/tool are described in the following sections in brief.

4.1.1. **VISUAL MILIEU AND APPLICATION DEVICES**

The GUI, as a revolutionary package, becomes enhanced in the micro-computer application. It, indeed, is widely accepted that developing a window-based application is more complex and dynamic rather than other its applications, viz. Disk Operating System (DOS). By nature, the package developed on the visual platform requires some degree of knowledge and expertise. The VB has been selected for the development of the GUI for the thesaurus because of the following reasons:

1. It is easier in programming, as a set of independent codes are used to activate for easy recognition and consequent responses.
2. It has the Project system rather than the programme.

Although it is not necessary to introduce the package - Visual Basic - in detail, because the main purpose of research is to the develop a thesaurus model, but as per requirement a minimum clear picture about it, in terms of its application devices, will be presented here for very basic understanding of what has been created.

The VB menu bar includes three things : title bar, a minimized button, and maximize button, a control menu, and the name of menus which can be used to activate the respective window. Although the menu location has set in the programme itself but it can also resize and move any where on the screen as an extra facilitation. There are certain procedures having to follow during the application of menu bar. They are given as follows :

- To activate the window in order to open menu and display its commands one can follow the steps of either by clicking menu name or defined shortcut access key.

- The special feature of commands is that it is followed by the ellipsis which is assigned by the dots for display a dialog box containing the respective information.

- Toolbar is set up with the buttons used as a shortcut to the menu commands. It contains the following buttons given from left to right manner. They are : Add Standard EXEProject, Add Form, Menu Editor, Open Project, Save Project. Cut, Paste, Find, Can't Undo, Can't Redo, Start, Break, End, Project Explorer, Properties Window, Form Layout Window, Object Browser, and Tool Box.

Tool Box is a collection of tools which are used to develop the GUI application. There are twenty one tools available in the Tool Box having different functions. From application point of view, they can only be used in the open form of the project being developed. There are the facility of choosing and closing the Toolbox. To open it, choose Toolbox from the Window menu and to close. double-click the Toolbox's Control menu.

Form window functions as a canvas of VB. It plays a crucial role in preparing the Visual Interface (i.e. GUI) because the end-user directly interacts with the form which is used to create different types of windows. The information, as an output, displays on the form through the respective boxes made on the form. A brief account of the description and functional utility are given below :

- It is a window-format display area which is seen at the application time.
- It is used to create not only the window but also the dialogbox in the application.
- Application of controls is drawn and activated on the Form Window.
- At the opening time of New Project, it dispalys empty with Form 1 title.
- The application components : **objects** and **controls** are used to design and develop the visual application.

Properties Window function as an attributive setting which controls the activate Object. It contains a list of properties or elements of the selected Object - form or control. The properties can be changed at the time of design and development of the application. The setting of properties functions as the current setup of run-time application. Its features and applicational utility are given below :

- Properties Window contains a list of properties of the activate object.
- The properties are arranged in alphabetic order.
- The properties function as attributive elements of the respective object which has been selected.
- By nature, each property can be changed under the possible nature and range as given as its values.
- Properties Window shows the current setting of property.
- Only one property can be changed at a time.
- More than one control can be accessed at a time alnogwith its common properties window.

It is a window-environment where the programming codes are written, displayed and modified. Each object is associated with a group of procedures. Those procedures are executed at the run-time. The features and functions are given below.

- There is interrelationship between the object and procedures.
- The procedure(s) is/are activated at run-time.
- A procedure is a group of statements in the VB programming language.
- It carries the action-oriented as well as event-oriented functional procedure. By name it is called the event procedure.
- More than one window can be opened and accessed at a time in order to edit the programme-codes.
- It is faciliated with the cut-paste-copy from one form window to another.

The Debug window is used to see the expression at run time. Its features, components and functions are summerised in terms of Debug Window and Watch Window as shown below:

- It is used to debug the codes.
- This window is activated only at run-time.
- It opens automatically at run-time.
- In order to test the codes line by line, it can be accessed at the break mode.
- It is not possibile to enter into the statement at run-time.
- A selected expression values can be seen.

A menu is created in order to make the package customised. Thus it is used to make the customised menu along with its defined properties and values. In order to open the Menu Design

Window, select a form on which the menu can be designed and then select the Menu design item from Window Menu. The description about the components is given below:

- It is a text box where the name of menu or commands is entered.
- The entered names appear on the menu or menu bar.
- In order to create a separate bar in menu, type single hyphen, i.e. (-).
- For keyboard access to the menu item, type the mark (&) before the letter. It appears in underscore mode which makes the user to access the command by pressing 'Alt' plus the underscored letter at the run-time.

4.2. GUI: *DESIGN AND DEVELOPMENT*

The visual intraction between the system and user becomes the primary requirement of application here. In the same way, as per the current thesaurus model, nineteen windows are designed and displayed, as they are shown in the Section 4.2.2. They are based on the following visual application-strategies:

- Scheme Planning : as a user requires to see on the display modes: Body and Information
- Menu-faciliation modes - commond button, option button.
- Range of the commands and options-events and controls.
- Windows resizing facility - fix mode.
- Each window has the associative information window.
- Interlink among windows- as shown in the section 4.2.2.
- Visual Basic 5.0 has been used do make the GUI for the current thesaurus.

4.2.1. THES. MAK PROJECT : *A PROPOSED THESAURUS MODEL*

The following processes are adopted for developing and desigining the application. It may be noted that instead of preparing the menu, we have created the command buttons which enable the user to interact faster than selecting the commands from menu. This phase of process consists of three steps : creating a New Project, creating a Form for each window, and setting the controls on the window.

4.2.1.1. DESIGN AND DEVELOPMENT PROCESS

A New Project : For creating a New Project as a part of visual application follow the following steps:

Steps :

1. Select the New Project command from the File Menu.
2. Create the form(s) and module(s).
3. Save all open files in it.

Features :

- All files created within a project can be saved and stored in it.
- Only one project can be opened at a time.
- In order to specify the default setting, modify the Autoload.mak project.

Creating Form : The steps followed in creating the form alongwith the features are given below :

Steps :

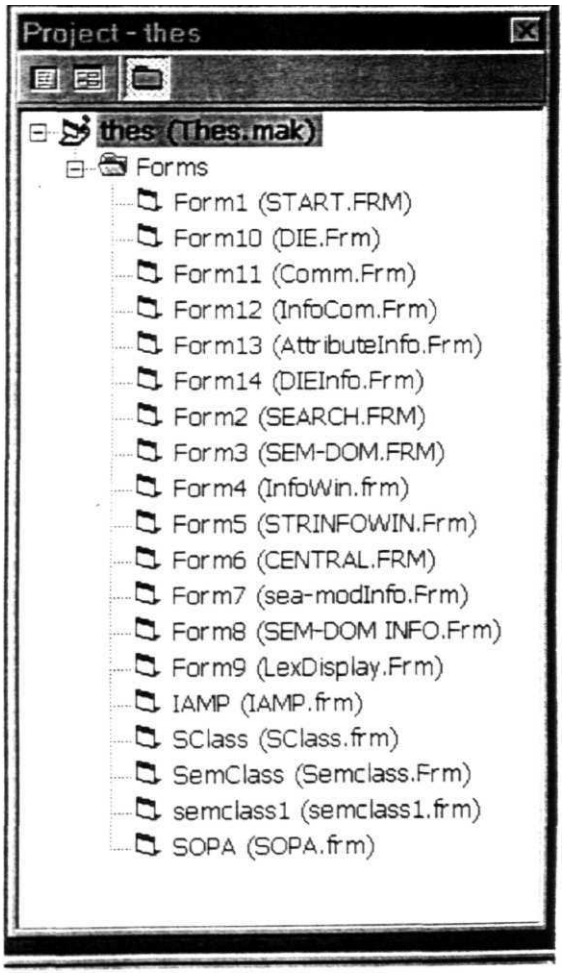
1. Select the New Form command from the File Menu.
2. Click the selected command or press Enter button.

3. You can save it by saving the current Project.
4. Select the Insert menu to add/insert the new form in the project.

Features:

- The first form displays automatically at the VB run-time.
- It functions in the window environment.
- The controls can be drawn and one can edit on it.
- It can be minimized and maximized by the Control Menu.

A Window of the **thes.mak** Project is shown below. It consists of nineteen forms created for the GUI of the present project of Hindi electronic thesaurus.



Thes.Mak Project

- **Control Setting** : The Toolbox consists of **the** control tools which are used to develop the GUI. **In order to use the** Tool Box, **the** designer **and** developer has to follow the following steps :

Steps :

1. Select the appropriate tool from the Tool Box by clicking it.
2. Drag it over the Form Window.
3. Draw it as per necessity.

Features :

- It has standard VB tools as well custom-made tools.
- It can be moved anywhere on the form.

4.2.1.2. **PROPERTIES** SETTING

Properties are the attributive elements of the form and control. There are a specific collection of properties for the form and each control. They are characterized with the object's features, viz. caption, style, size etc. and the it function, viz. enabled or not, true or false, etc. The properties can be set either at the design time in the Properties Window or at the run time in the Code Window.

4.2.13. **CODE WRITING**

The code is entered in the Code Window as a part of programme writing.The Code Window is faciliated with the automatic formatting syntax checking capability. To display the syntax error, set the syntax checker on. Then it will display the message regarding syntax error. The code setting can be matched by setting the event procedure and general procedure.

The important step in the application development is debugging of the errors which can be of three types - Compile errors, Run-time errors, and Logical errors. The Compile errors occur in the following cases :

- If the code is incorrectly constructed.
- If there is a mismatched control structure.
- In case of programming faults - wrong spelling, missing seperator, wrong syntax representation, unrecognized punctuation, etc.

The run-time error is product of illegal operation. It occurs at the execution time. The logical errors occur with unintended result. The programme does not perform the correct result as expected. It happens due to the inappropriate syntax selection or some wrongly recognized arguments. These errors can be detected by debugging the application by using the Debug tools.

4.2.1.4. MAKING EXE FILE

Once the whole process has been completed for application, creation of executable file is required to execute the application for output. It can be created by selecting the Make EXE File command from the File Menu. Here, the forms and modules of a project are assembled into single executable file. It is created automatically.

4.2.2. WINDOW-NETWORK

The Thes.Mak Project consists of 19 forms. They can be divided into two groups - body part and information part. The interlink between them is shown in the following diagram:

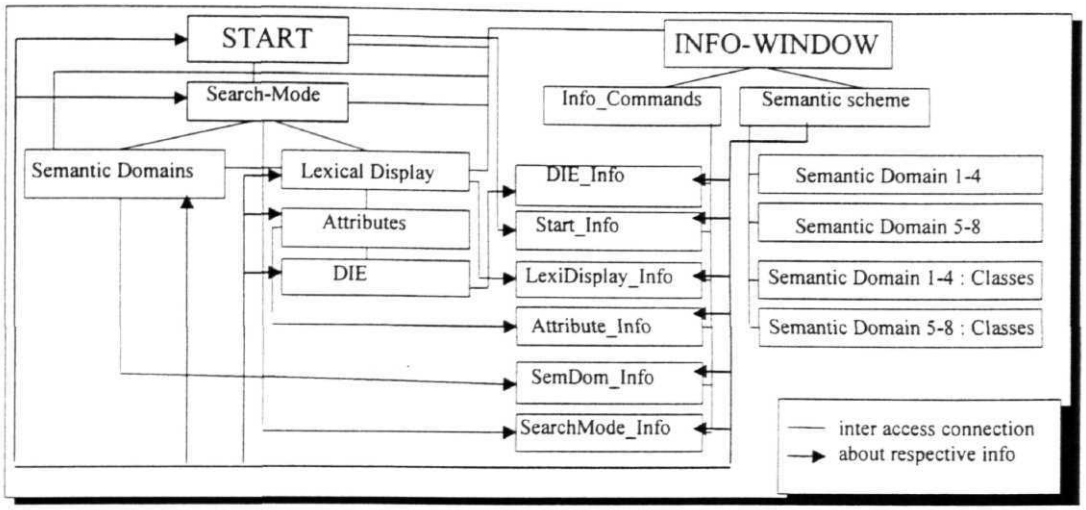
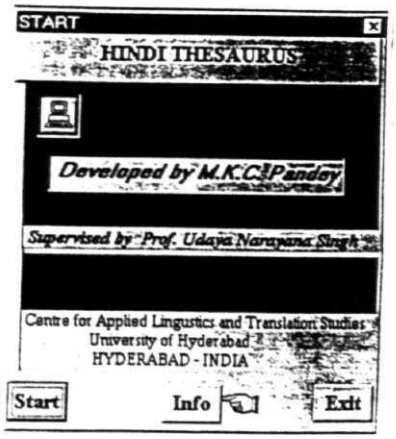


Diagram 11 : Showing Interlink among Forms

The application developed can be divided into two parts of processes. They are the GUI as output-display where there is interaction between the user interfaces the computer and internal in-built processes: the whole scheme (project) is setup in the unique and executable form. The interrelation between the input and output should be user-oriented. For applicational development on the visual environment, one should create the visual user interface. In order to create the user interface, it is necessary to set the properties for each form and control and to write the code. Thus the whole process of creating visual application can be divided into two categories : *Creating* the GUI and *Executing* the programme. The nature and functions, in terms of screen display, introductory information about the window, consisting commands with their functions, and important algorithm about the programming (as instances), of each form are explained below.

4.2.2.1. START FORM : *FORM 1 : START*

The Start form functions as a gateway of thesaurus. It consists of the detail about the registration and title. This is the first form which appears at the EXE run-time. Its screen display is shown below :



Screen Display : Start Form

It consists of the three command-functions Start, Info, and Exit. The functions of these commands are given below :

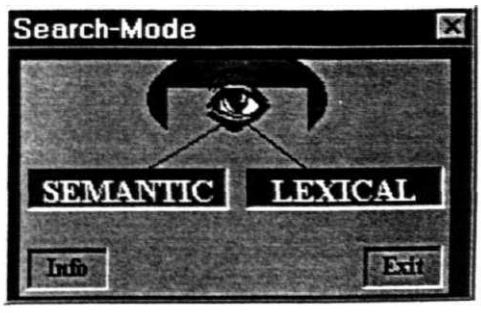
Buttons	Applications
Start	* Run the thesaurus. * Go to the Search_Mode form
Info	* Display the Info Window.
Exit	* Close the running Application.

From the programme point of view, it is based on the Visible functions. It mainly consists of how one can from one screen to another and how the other form can be open. The sample programme about the Start command has been shown below:

```
Private Sub START_Click()
        'Display Search_form
        Form2.Visible = 1
        Form1.Visible = 1
End Sub
```

4.2.2.2. SEARCH-MODE FORM : *FORM 2 : Search-Mode*

The Search-Mode form functions as a double-gateway. Here, the user is facilitated with the two ways accessing facility, i.e. semantic and lexical. He can access the thesaurus either from the semantic or from the lexical mode, depending on whether one would like to find the synonyms and related word directly or would explore into meaning shades in order to find the right word. The screen display has been given below :



Screen Display : Search-mode

This form consists of the four commands namely SEMANTIC. LEXICAL. Info, and Exit. The functions of these commands are given below :

Button	Application
SEMANTIC	* Open the Semantic Domain form.
LEXICAL	* Open the Lexical Display form.
Info	* Display the Info_Window.
Exit	* Close the running Application.

From the programming point of view it deals with the visible property-value. Moreover, in the case of LEXICAL command, the main functions open the Lexical Display form and set the focus on the HWEditbox. Some programmes are shown below as instances :

```
Private Sub lexical_Click()
        'Display Form5'
        Form6.Visible = 1
        Form2.Visible = 0
        Form1.Visible = 0
        Form6.HWEditbox.SetFocus
End Sub

Private Sub Semantic_Click()
        'Display Scr.Frm 3'
        Form3.Visible = 1
        Form2.Visible = 0
        Form1.Visible = 0
End Sub
```

4.2.23. SEMANTIC DOMAIN FORM : *FORM 3 : Semantic Domains*

This is one of the main screens of the present thesaurus. It includes three major components : Semantic Domains, Word-List Box and Semantic Classes. The feature and function of these components are given below:

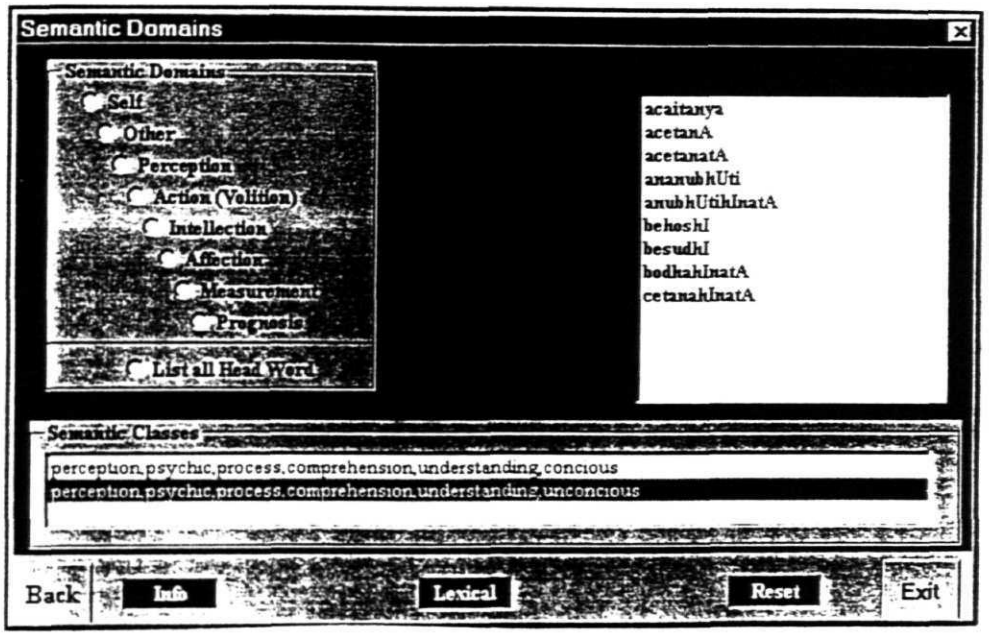
- **Semantic Domains** : It has been divided into two parts. The first part consists of eight semantic domains covering world-knowledge. This eight-part scheme is the result of a hypothesis which is. of course, open to falsification or modification. They are Self, Other, Perception, Action (Volition). Intellection, Affection, Measurement, and Prognosis, whereas the second part consists of the List of All Head Words. They are placed in the optional mode.

- **Semantic Classes** : This list box is used to display the semantic classes of the selected Semantic Domain. For instance, if one selects the Semantic Domain **'Seir** from the list of domains and clicks its option button, its semantic class

will get displayed in this box. The semantic classes are listed in alphabetical **order. Here,** the user can select any one of the semantic classes and click it once, in which case the headword coming under this class will be displayed in the Word-List box.

- **Word-List** : This list box is used to display the headword of thesaurus. All headwords are displayed in the alphabetical order. Here, the user has to follow the following steps to look the searching word :
    - Select the word.
    - Click twice on it.

  As soon as he clicks twice on the selected word, the word will copy in the **Headword** Editbox of the Lexical Display screen.



Screen Display : Semantic Domains

This form consists of three types of command-functions : Optional Button, Command Button, and box-command. The information about the nature and function of these commands are given below.

- **Option Button** : There are nine option buttons provided in this form. They perform three functions :
    - Only one option can be selected at a time.
    - Use a ***click*** to list the semantic classes of the selected semantic domain in the Semantic Classes box.
    - Use the ***double-click*** to display the headwords of the respective semantic domain in the Word-List box.

- **Command Button** : There are two command buttons used to go back and exit from the screen. Applications of these commands are given in the following table.

Box-Command : This form consists of three box-commands used to display the information, to go to the Lexical Display form, and to reset the current information. Their application has been shown in the table given below :

Button	**Application**
Back	* Go to the Search-Mode form.
Info	* Display the Info_Window.
Lexical	* Open the Lexical Display form.
Reset	* Reset the current application
Exit	* Close the running Application.

From the programming point of view, it deals with the following main points : How is the particular semantic domain matched with the lexical database (i.e. Lexis.mdb)? How does one retrieve the semantic classes and headwords, or reset the current display by clearing the boxes, moving from the current form to another form, listing the semantic classes and headwords in alphabetical order? Here, some of the programming samples are given as instances:

- The programme **Private Sub Action_Click()** is based on the following points :
    - Open the **Datal.DatabaseName** = **"thesdata\Lexis.mdb"**
    - Open to the **Datal.RecordSource=** "entry"
    - Refresh the current database, especially the table.
    - Display the semantic classes of the semantic domain '**Action**' in the Semantic Classes box.

```
Private Sub Action_Click()
        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SClListbox.Clear
        'Match ACTION
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
```

```
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "v" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SCIListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                                If Data1.Recordset(val3) = SCIListbox.List(j) Then
                                                        found = True
                                                        Exit For
                                                End If
                                        Next j
                                        If found = False Then
                                                SCIListbox.AddItem Data1.Recordset(val3)
                                                lIndex = lIndex + 1
                                        End If
                                End If
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        Action.Value = False
        Action.Enabled = True
End Sub
```

The programme **Private Sub Action_DbCIick()** is based on the following points :

- Open the **Datal.DatabaseName = "thesdata\Lexis.mdb"**
- Open to the **Datal.RecordSource = "entry"**
- Refresh the current database, especially table.
- Display the headwords of the semantic domain **'Action**'in the Word-List box.

```
Private Sub Action_DblClick()
        'Clear ListBoxes
        SCIListbox.Clear
        SCIListbox.Clear
```

```
        'Match Action
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If  Data1.Recordset(val3) = "v" Then
                                found = True
                                WordListbox.AddItem Data1.Recordset("hw")
                        End If
                        If found = True Then
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub
```

The programme **Private Sub Back_CIick()** plays the two main functions :

- Clear the boxes (WordListbox and SClListbox).
- Display the form no. 1 and form no. 2 (i.e, START and SEARCH-MODE forms respectively)

```
Private Sub Back_Click()
        'Clear the boxes
        WordListbox.Clear
        SClListbox.Clear
        'Display Scr.Frm2'
        Form3.Visible = 0
        Form1.Visible = 1
        Form2.Visible = 1
End Sub
```

The programme **Private Sub ListHW_DblClick()** deals with following points

- Clear the boxes : WordListbox and SCiListbox/
- Open the **Datal.DatabaseName** = **"thesdata\Lexis.mdb"**
- Open to the **Datal.RecordSource** = **"entry"**
- Refresh the current database, especially the table.
- Display the all headwords in the Word-List box.

```
Private Sub ListHW_DblClick()
        'Clear ListBoxes
         WordListbox.Clear
        SCIListbox.Clear
        Dim intWctr As Integer

        intWctr = 0
        'set database
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        Do While Data1.Recordset.EOF = False
                If Data1.Recordset.Fields("hw") <> "" Then
                        WordListbox.AddItem Data1.Recordset("hw")
                        intWctr = intWctr + 1
                End If
                Data1.Recordset.MoveNext
        Loop
        Text1.Text = intWctr
End Sub
```

The **Private Sub Reset_CIick()** programme is used to clear the WordListbox and SCiListbox. This small programme could be shown below :

```
Private Sub Reset_Click()
        WordListbox.Clear
        SCIListbox.Clear
End Sub
```

The programme **Private Sub SCIListboxDblClickO** deals with following points :

- Clear the WordListbox.
- Open the **Datal.DatabaseName = "thesdataVLexis.mdb"**
- Open to the **Datal.RecordSource = "entry"**
- Refresh the current database and table.
- Display the all headwords of the selected semantic classes in the Word-List box.

```
Private Sub SClListbox_DblClick()
        WordListbox.Clear

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val As String
        val1 = "sc"
        Do While Data1.Recordset.EOF = False
                For i = 1 To 12 Step 1
                        val2 = i
                        val3 = CStr(val2)
                        val4 = val1 + val3
                        If SClListbox.List(SClListbox.ListIndex) = Data1.Recordset(val4) Then
                                WordListbox.AddItem Data1.Recordset("hw")
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub
```

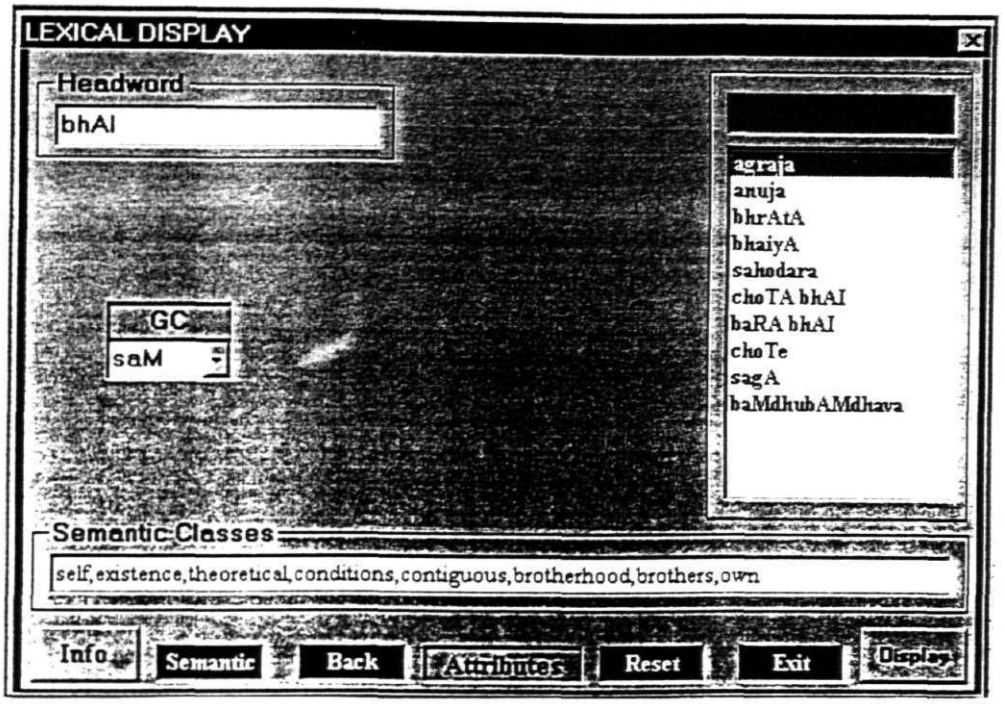
The functions of programme **Private Sub WordListbox_Click()** are as follows :

- Open the form 6 (i.e. the Lexical Display form).
- Copy the selected word into the HWEditbox of the Lexical Display form. This action is carried out by clicking the selected word in the WordListBox.

```
Private Sub WordListbox_DblClick()
        If  WordListbox.List(WordListbox.Listlndex)  o "0" Then
                Form6.Visible = 1
                Form3.Visible = 0
                Form6.HWEditbox.Text = WordListbox.List(WordListbox.Listlndex)
        End If
End Sub
```

4.2.2.4. LEXICAL DISPLAY FORM : *FORM 6 : LEXICAL DISPLA Y*

The Lexical Display is the main form of the Thes.Mak project. It consists of four components: Headword, GC, Display box and Semantic Classes. The nature and functions of these components are given below along with their screen display :



Screen Display : Lexical Display

Headword : It is an edit textbox in which the searching word is typed. There are two ways to enter the headword in this box.

- One can type the word which one wants to search.
- One can select the word from the Word-List box of the Semantic Domain form and click on it. Then the word will be automatically copied into this box.

GC : It is a Combobox. In this the grammatical category (or, categories) of the selected word appears. Here, by pressing the combo button, one can see a list of possible grammatical classes in respect of the searched word. This application is based on the value of HWEditbox. Let us see a search of noun as a part of programme which is defined as "saMT in the following programming :

```
Private Sub GCcombox_Click()
        Dim val As String
        val = HWEditbox.Text

        'Clear Boxes
        SynListbox.Clear
```

```
        LSClistbox.Text = ""
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        'GCcombox noun display
        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        found = False
                        Dim val1 As String
                        Dim val2 As Integer
                        Dim val3 As String
                        Dim val4 As String
                        val1 = "sn"
                        If GCcombox.Text = "saM" Then
                                For i = 1 To 12 Step 1
                                        val2 = i
                                        val4 = CStr(val2)
                                        val3 = val1 + val4
                                        If Data1.Recordset(val3) <> "" Then
                                                found = True
                                                SynListbox.AddItem Data1.Recordset(val3)
                                        End If
                                Next i
                        End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If
                .
                .
                .
        Loop
End Sub
```

With this programme, the system will perform the following functions :

It will match the value of the Headword box.
It will clear both boxes : SynListbox and LSClistbox
It will set the database active with refresh.
it will match the word typed or entered into the Headword box with the fields namelv hw, iw, and sv. As soon as it gets the match, it will begin looking for the match with 'saM' within the *GC* box which stands for noun.
Then it will start searching right from field *snT to field 'sn12' and will display all the words found in the SynListbox..

Likewise, it will perform the function for other three grammatical classes, namely, Verb, Adjective, and Adverb.

Display Box : It is a SynListbox. It has two parts. In the first part (i.e. Titletext), the titles are displayed depending upon the information being searched. For instance, If one wants to see the antonyms of the searching word, the 'Antonyms' will appear in this box. Here, two main functions are performed. The first is performed with the programme of **Private Sub SynListbox_CIick()** as shown in the following sample programme of Noun Matching SD-SC:

```
Private Sub SynListbox_Click()
        LSClistbox.Text = ""
        Noun Matching SD-SC
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "sn"
        If GCcombox.Text = "saM" Then
                For i = 1 To 4 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val3) Then
                                LSClistbox.Text = Data1.Recordset("sc1")
                        End If
                 Next i
        End If
        Dim val5 As String
        Dim val6 As Integer
        Dim val7 As String
        Dim val8 As String
        val5 = "sn"
        If GCcombox.Text = "saM" Then
                For j = 5 To 8 Step 1

                        val6 = j
                        val7 = CStr(val6)
                        val8 = val5 + val7
                         If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val8) then
                                LSClistbox.Text = Data1.Recordset("sc2")
                         End If
                Next j
        End If
        Dim val9 As String
        Dim val10 As Integer
        Dim val11 As String
        Dim val12 As String
```

```
val9 = "sn"
If GCcombox.Text = "saM" Then
        For k = 9 To 12 Step 1
            val10 = k
            val11 = CStr(val10)
            val12 = val9 + val11
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val12) Then
                    LSClistbox.Text = Data1.Recordset("sc3")
            End If
                Next k
End If
.
.
.
```

- This programme performs the following functions :

    - As soon as one clicks at the highligted word (i.e. synonym), its semantic class will appear in the Semantic Class box.
    - The system will begin to search 'saM' (Noun) from Field snl to Field snl2 depending uopn the match with the GC box. Here particular performance occurs in three slots :

        - In the first slot, the system will display the SCI (i.e. Semantic Class no. 1) for the snl to sn4.
        - In the second slot, the system will display the SC2 (i.e. Semantic Class no. 2) for the sn5 to sn8.
        - In the third slot, the system will display the SC3 (i.e. Semantic Class no. 3) for the sn9 to snl2.

    - Likewise, it will perform for the other three grammatical categories, too, and their respective fields : from 13sn to 48sn for the sc5 to sc 12.

The second function is performed with the **Private Sub SynListbox_DblCIick()** programme. This small programme is used for the cross-reference. If one wants to see the cross references of the word being searched, one can click twice at the highlighted word. As soon as one clicks twice, the word will be copied in the Headword box. With the help of a Display button, one can then see all the possible information about it. The programming is given below :

```
Private Sub SynListbox_DblClick()
        LSClistbox.Text = ""
        HWEditbox.Text = SynListbox.List(SynListbox.Listlndex)
End Sub
```

This form consists of seven commands. Their applications have been shown in the table below

Button	Application
Info	* Display the InfoWindow.
Semantic	* Open the Semantic Domain form
Reset	* Clear all the boxes.
Back	* Go to the Search-Mode form.
Exit	* Close the running Application.
Display	* Display the grammatical category in the GC box. * Display the title in the LSClistbox.
Attributes	* Open the Attributes form.

The performance of Display button depends upon the type entered of the searching word in the Headword box. Only if there is a word, it will be activated and display the grammatical classes of that word in the GC combobox and the title Synonyms in the LSClistbox. For this performance, the system will do the following tasks as shown in the following programme :

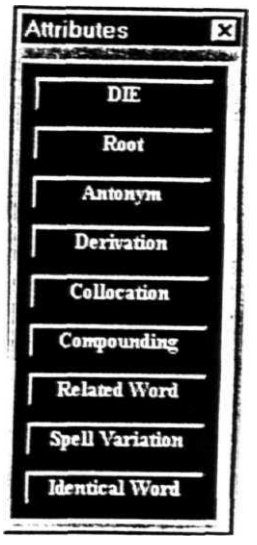
- It will clear the all boxes, namely, SynListbox GCcombox and LSClistbox.
- It will open the database and its respective table from the refresh point.
- It will start the search from the Field gc1 to gc4. If it gets any text filled up in these fields, it will display in the GC box and at the same time it will also display the Synonyms in the Title box of the LSClistbox.

```
Private Sub Display_Click()
        Dim val As String
        val = HWEditbox.Text

        'Clear list box/comboboxh
        SynListbox.Clear
        GCcombox.Clear
        LSClistbox.Text = ""
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
```

```
        Dim val4 As String
        val1 = "gc"

        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                        Data1.Recordset("iw") Then
                        For i = 1 To 4 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If val <> "" Then
                                        If Data1.Recordset(val3) <> "" Then
                                                found = True
                                                GCcombox.AddItem Data1.Recordset(val3)
                                        End If
                                End If
                        Next i
                        Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If
        Loop
        Titletext.Text = "Synonym"
End Sub
```

4.2.2.5. ATTRIBUTES FORM : *FORM 11: Attributes*

The attributive form is a form of commands, it consists of the nine text commands : DIE, Root, Antonym, Derivation, Collocation, Compounding, Related Word, Spell Variation, and Identical word. Its screen display, nature and functions are given below :



Screen Display : Attributes

DIE : This command opens the DIE form. It can be opened only if the Headword box (HWEditbox) has the appropriate word. Its programming function has been shown in the following **Private Sub DIE_CIick()** programme.

```
Private Sub DIE_Click()
        Form10.DIEListbox.Clear

        Dim val As String
        val = Form6.HWEditbox.Text

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        Data1.Recordset.MoveFirst
        Do While Data1.Recordset.EOF = False
                If Form6.HWEditbox.Text = Data1.Recordset("hw") Then
                        Form10.Visible = 1
                        Exit Do
                Else
                        Form10.Visible = 0
                End If
                Data1.Recordset.MoveNext
        Loop
End Sub
```

- **Root**: With a click, it will show the root of the searching word. This root will appear in the lower part of LSClistbox of Lexical Display form and in the upper part the the title "Root' will appear.

- **Antonyms** : With a click, the antonyms of the searching word will appear in the lower part of LSClistbox of Lexical Display form and in the upper part the title Antonyms will appear. There is facility of providing 12 antonyms in this thesaurus based on the assumption of matching with 12 different semantic domains/classes. But there is also a facility that all these 12 antonyms can be shown only in the case of one or more than one semantic domain or class. The interlink of search of these 12 antonyms can be identified in the following **Private Sub antoClickQ** programme :

```
Private Sub anto_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        Rem Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""
```

```
        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "ant"
        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
                If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        For i = 1 To 12 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data2.Recordset(val3) <> "" Then
                                        found = True
                                        Form6.SynListbox.AddItem Data2.Recordset(val3)
                                End If
                        Next i
                        If Not found Then
                                MsgBox "Antonym is not available", vbOKOnly
                        End If
                        Exit Do
                End If
                If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <>
                Data1.Recordset("iw") Then
                        Data1.Recordset.MoveNext
                        Data2.Recordset.MoveNext
                End If
        Loop
        Form6.Titletext.Text = "Antonym"
End Sub
```

With a click on the respective buttons, namely. **Derivation, Collocation. Compounding,** and **Related Word,** the related information will appear in the lower part of LSClistbox of Lexical Display form . The algorithm follwed here is same as for antonym.

With a click on the **Spelling Variation** and **Identical Word** buttons, the related information will appear in the lower part of LSClistbox of Lexical Display form. The algorithms followed in both cases are:

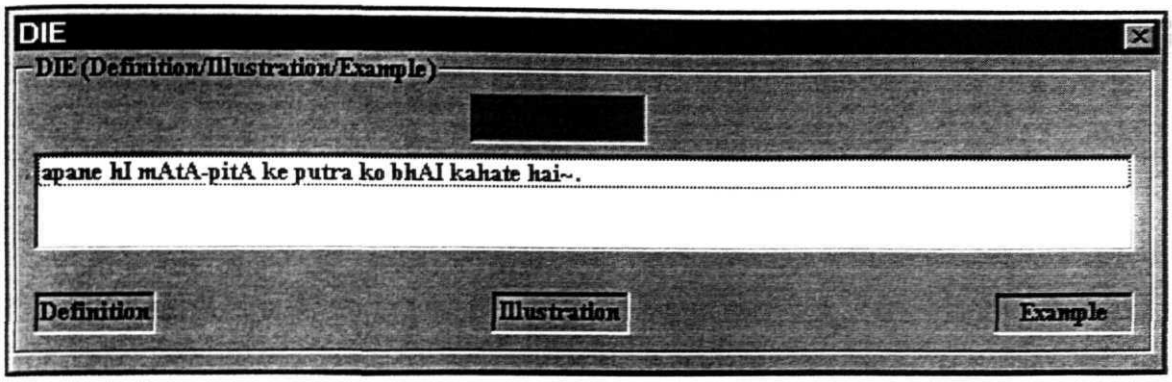
- Spelling Variation = Headword and Identical Word
- Identical Word = Headword and Spelling Variation.

```
Private Sub spllvar_Click()
        Dim val As String
        val = Form6.HWEditbox.Text
        'Clear list box

        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Data1.Recordset.MoveFirst
        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Then
                    If Data1.Recordset("sv1") <> "" And val = Data1.Recordset("hw") Then
                        found = True
                        Form6.SynListbox.AddItem Data1.Recordset("sv1")
                    End If
                    If Data1.Recordset("sv1") <> "" And val <> Data1.Recordset("hw")  then
                        found = True
                        Form6.SynListbox.AddItem Data1.Recordset("hw")
                    End If
                    Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If

        Loop
        If Not found Then
                MsgBox "Spelling Variation is not available", vbOKOnly
        End If
        Form6.Titletext.Text = "Spell Variation"
End Sub
```

- In above programme, the main fact taken into account is that the spelling variation is equal to having an additional headword. Therefore, the information of the head word regarding all attributes should be displayed in the spelling variation cases also. For instance, the synonyms of the headword (a searching word) will be the synonyms of its spelling variation word. This is also true for Identical word.

4.2.2.6. DIE FORM : *FORM 10 : DIE*

The word DIE stands for the Definition, Illustration, and Example. This form is basically a part of the Lexical display form. It is open from the Attribute form which functions either as an intermediator or as the command form of the Lexical Display. It has two components : **Title text box.** and a **List box.** In the title box, the titles : **Definition, Illustration,** and **Example** appear, whereas the information about them is shown separately. The screen display has been shown below :



Screen Display : DIE form

Definition : The present thesaurus provides the facility of an explicit and unique definition. Here the explicit and unique refer to the form of definition. The limitation of this thesaurus is that it provides only one definition. From the programming point of view, the main function of the definition is to look for the searching word typed into the Headword box of the Lexical Display form and if it is found, display its (definition) into the Listbox of the DIE form. The whole process has been shown in the following **Private Sub defination ClickQ** programme:

```
Private Sub defination_Click()
        Dim val As String
        Dim found As Boolean
        Dim blank As Boolean

        val = Form6.HWEditbox.Text

        DIEListbox.Clear

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "ide"
        Data1.Refresh

        found = False
        Data1.Recordset.MoveFirst
        Do While Data1.Recordset.EOF = False
            If Trim(val) = Data1.Recordset("hw") Or Trim(val) =
            Form6.Data1.Recordset("sv1") Or Trim(val) = Form6.Data1.Recordset("iw") Then
                        found = True
                        If Data1.Recordset.Fields("defin") <> "" Then
                                blank = False
                        Else
                                blank = True
                        End If
                        If found Then
                                If blank Then
```

```
                    MsgBox "Definition is not available"
                    Exit Do
                Else
                    DIEListbox.AddItem Data1.Recordset.Fields("defin")
                    End If
                End If
            End If
            Data1.Recordset.MoveNext
        Loop
        DIETitlebox.Text = "Definition"
End Sub
```

- **Illustration** and **Example :** The present thesaurus is also facilitated with the example(s). There are possibilities of providing maximum 12 examples and illustration information depending upon the context in which the searching word is used. The function and retrieve processes have been shown in the following **Private Sub Example_Click()** programme. The conception followed here is applicable for the Illustration, too.

```
Private Sub Example_Click()
        Dim val As String
        val = Form6.HWEditbox.Text
        DIEListbox.Clear
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "ide"
        Data1.Refresh
        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "examp"

        Do While Data1.Recordset.EOF = False
            If val = Data1.Recordset("hw") Or val = Form6.Data1.Recordset("sv1") Or val =
            Form6.Data1.Recordset("iw") Then
                For i = 1 To 12 Step 1
                    val2 = i
                    val4 = CStr(val2)
                    val3 = val1 + val4
                    If Data1.Recordset(val3) <> "" Then
                        If val = Data1.Recordset("hw") Then
                            found = True
                            DIEListbox.AddItem Data1.Recordset(val3)
                        End If
                    End If
```

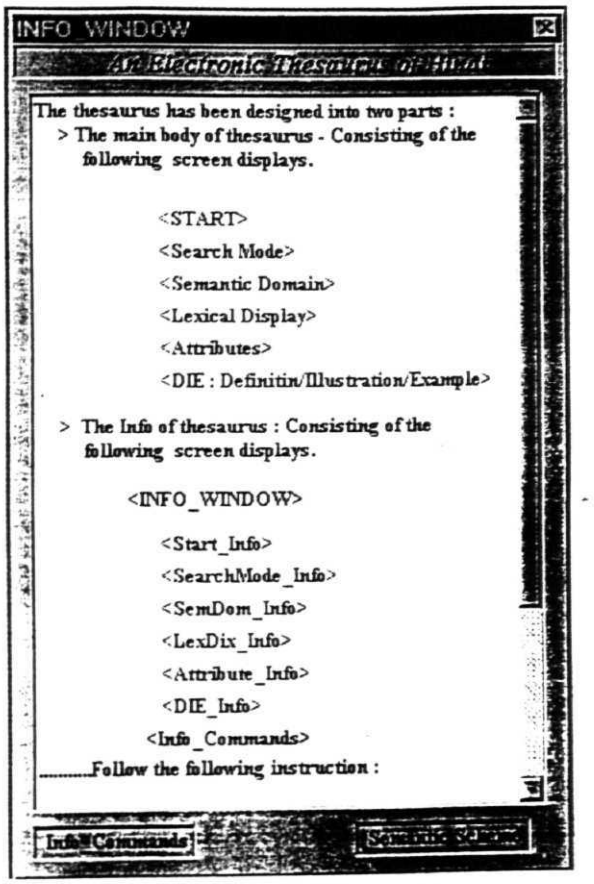
```
                    Next i
                    If Not found Then
                            MsgBox "Example is not available", vbOKOnly
                    End If
                    If val <> Data1.Recordset("hw") Then
                            SynListbox.AddItem Data1.Recordset("hw")
                    End If

                    Exit Do
            End If
            If val <> Data1.Recordset("hw") Then
                    Data1.Recordset.MoveNext
            End If
        Loop
        DIETitlebox.Text = "Example"
End Sub
```

This form has four commands - Definition, Illustration, Example and Exit. Their applications are given in the following table :

Buttons	**Applications**
Definition	* Display the definition of the headword in the List box. * Display the title 'Definition' in the small text box.
Illustration	* Display the illustration of the headword in the List box. * Display the title illustration" in the small text box.
Example	* Display the example of the headword in the List box. * Display the title 'Example' in the small text box.
Exit	* Close the running Application.

4.2.2.7. INFOJWINDOW FORM : *FORM 4 : INFOJVINDOW*

The Info_ window consists of the information about the present thesaurus in terms of its different screen displays and some relevant information. It is noted here that the information given here is represented in the points format. The Info_window form and the information are shown below :



Screen Dispaly : Info_Window Form

The thesaurus has been designed into two parts :

> The main body of thesaurus - consisting of the following screen displays:

- » START
- » Search-Mode
- » Semantic Domain
- » Lexical Display
- » Attributes
- » DIE (definition /Illustration/Example)

> The Info of thesaurus - consisting of the following screen displays:

- » INFO_WINDOW
- » Startlnfo
- » SearchMode_Info
- » SemDomlnfo
- » LexDisInfo
- » Attributelnfo
- » **DIEJnfo**
- » InfoCommands
- » Semantic Scheme

```
        >> Semantic Domain 1-4
        >> Semantic Domain 5-8
        >> Semantic Domain 1-4 : Classes
        >> Semantic Domain 5-8 : Classes

.....Follow the following instruction :

        @ For further information select the respective
          Info_combox and click.

.....NOTE :

        # The Info_Command window consists of the
          commands for  other Info windows.
```

This form consists of the folowing commands :

Buttons	**Applications**
Info_Commands	* Display the Info_Commands form.
Semantic Scheme	* Display Semantic Scheme form.

4.2.2.8. INFO_COMMANDS FORM : *FORM 12 : InfoJCommands*

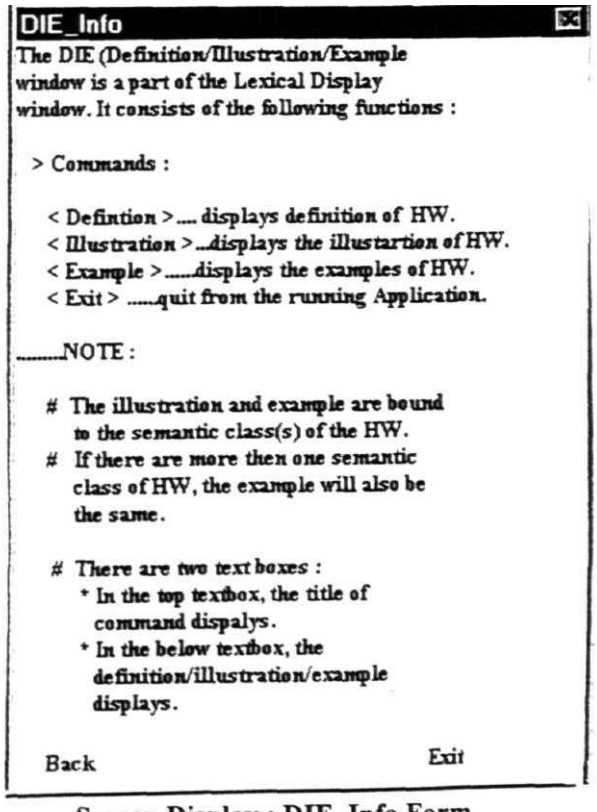
This form consist of seven box commands. They are DIE_Info, Start_Info, Lexic_Info, Attribute_Info, SemDom_Info, SearchMode_Info, and Exit. The screen display of this form has been given below which is followed by the application of commands :

Screen Display : InfoCommands Form

Buttons	Applications
DIE_info	* Display the DIE_Info form.
Start_Info	* Display the Start_Info form.
Lexic_Info	* Display the Lexic Info form.
Attribute_Info	* Display the Attribute_Info form.
SemDomInfo	* Display the SemDom_Info form.
SearchMode_Info	* Display the SearchMode_Info form.
Exit	* Close the running Application.

4.2.2.9. DIE_INFO FORM : *FORM 14: DIEJnfo*

This form is mainly related to the information about the DIE form. It gives all account of the function of DIE form, including the commands. The screen display, information, and the command **table** are given below :



Screen Display : DIE_Info Form

DIE-INFO WINDOW

The DIE (Definition/Illustration/Example) window is a part of the Lexical Display window. It consists of the following functions :

> Commands :

< Definition >.... displays definition of HW.
< Illustration >...displays the illustaration of HW.
< Example >.....displays the examples of HW.
< Exit > .....quit from the current window.

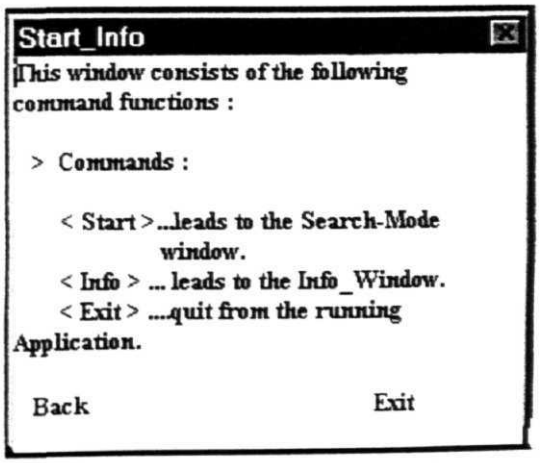
.....NOTE :

The illustration and example are bound to the semantic class(s) of the HW.
If there are more then one semantic class of HW, the example will also be of the same type.
There are two text boxes :
* In the top textbox, the title of command dispalys.
* In the textbox below, the definition/illustration/example displays.

Buttons	Applications
Back	* Go to the Info window form.
Exit	* Close the running Application.

4.2.2.10. STARTJNFO FORM : *FORMS: StartInfo*

The Start_Info form deals with the information about the Start form. It comprises the information about the Command functions . The Start_Info screen display alongwith its containing information, and the command tables are given below :



Screen Display : Start_Info Form

This window consists of the following command functions :

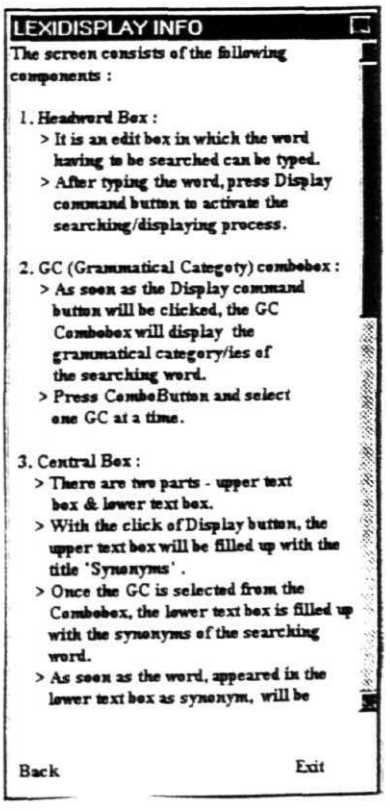
> Commands .

< Start >.. leads to the Search-Mode window.
< Info > ... .. leads to the Info Window.
<Exit> ... .. quit from the current window.

Buttons	Applications
Back	* Go to the Info_window form.
Exit	* Close the running Application.

4.2.2.11. LEXIDISPLAY INFO FORM : *FORM9 : LEXIDISPLAYINFO*

This form deals with the information about the Lexidisplay_Info form in terms of different components and command functions . The screen display of Lexidisplay_Info alongwith its information and commands are given below :



Screen Display : Lexidisplay_Info Form

The screen consists of the following components :

1. Headword Box :
   > It is an edit box in which the word having to be searched can be typed.
   > After typing the word, press Display command button to activate the searching/displaying process.

2. GC (Grammatical Category) combobox :
   > As soon as the Display command button will be clicked, the GC Combobox will display the grammatical category/ies of the searching word.
   > Press ComboButton and select one GC at a time.

3. Central Box :
   > There are two parts - upper text box & lower text box.
   > With the click of the Display button, the upper text box will be filled up with the title 'Synonyms' .

   > Once the GC is selected from the Combobox, the lower text box is filled up with the synonyms of the searching word.
      » As soon as the word, appearing in the lower text box as a synonym is shown, it will be marked/selected by an arrow, its Semantic Class automatically will display in the Semantic Classes Box.
      » With Double-Click on the word appeared in the lower box, it will be copied in the 'Headword' editbox for cross searching.
» Synonyms are listed according to the Semantic Classes group.
      » All information related to the searching word will be displayed in the lower box. Eg. Root, Antonyms ....
   > The upper box is filled up with respective info automatically. Eg. in case of Root, the Root text will appear in it and in the case of Antonym, the antonym text will appear, and so on and so forth.

4. Sematic Classes :
   > It consists of the display of Semantic Class of the selected synonym.

   > Only one Semantic Class will appear at a time.

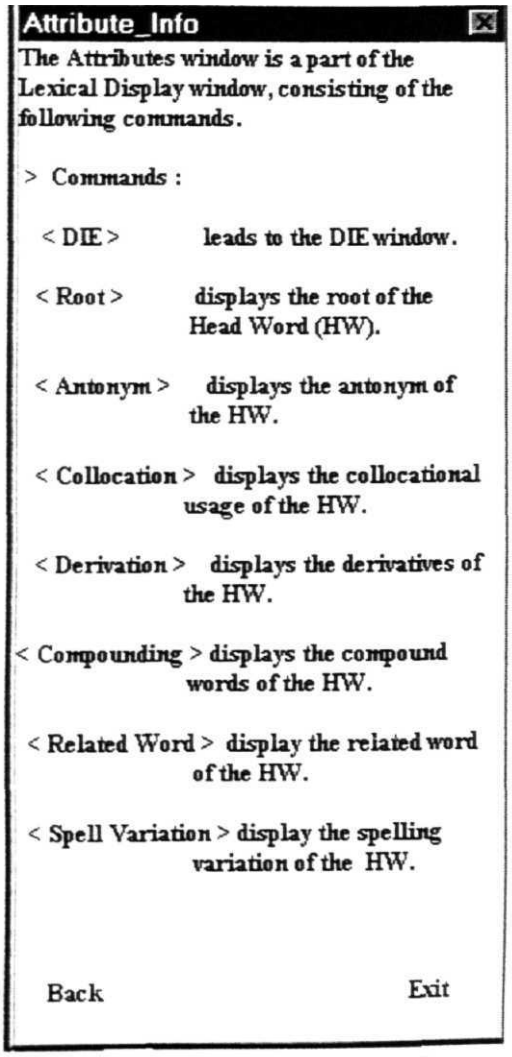
5. Commands :

   < Semantic >.....leads to the Semantic Domain screen.
   < Previous >.....set the previous operation by undoing the latest operation to it.
   < Reset >__makes every box empty for the reaccess ing operational process.
   < Back >.....leads to the Search-Mode screen.
   < Exit >.....quits from the current window.
   < Info >.....leads to the information window.
   < Display >.....makes the searching word activated.
   < Attributes >.....leads to the Attribute window.

Buttons	Applications
Back	* Go to the Info_window form.
Exit	* Close the running Application.

4.2.2.12. ATTRIBUTE_INFO FORM : FORM 13 : Attributejnfo

This Attributejnfo form embodies the information regarding the different commands used in the Attributes form. Its screen display, the embodied information, and commands are given **below:**



Screen Display : Attribute_Info Form

The Attributes window is a part of the Lexical Display window, consisting of the following commands.

> Commands :

<DIE>..... ...leads to the DIE window.

<Root>..... ...displays the root of the Head Word (HW).

< Antonym >.....displays the antonym of the HW.

< Collocation >..... ...displays the collocational usage of the HW.

< Derivation >.....displays the derivatives of the HW.

< Compounding >... ...displays the compound words of the HW.

< Related Word > .....displays the related word of the HW.

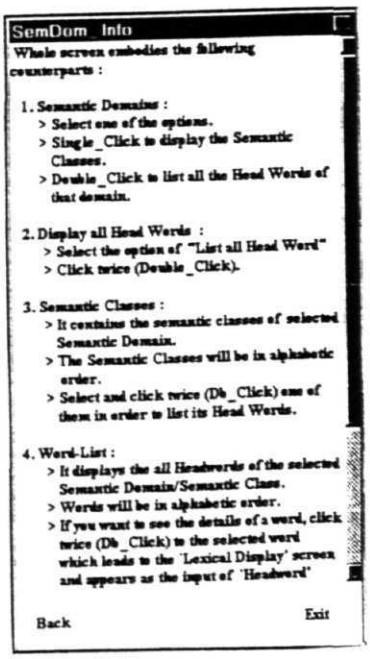
< Spell Variation > .... displays the spelling variation of the HW.

< Identical Word > . ...displays the identical wors of the HW.

Buttons	Applications
Back	* Go to the Info_window form.
Exit	* Close the running Application.

4.2.2.13. SEMDOMJNFO FORM : *FORM 8 : SemDom-Info*

The form Semdom_Info includes information about different components and commands used in this form. Its screen display, information, and commands are given below :



Screen Display : SemDom-Info Form

SEM-DOM INFO WINDOW

Whole screen embodies the following counterparts :

1. Semantic Domains :
   > Select one of the options.
   > Single_Click to display the Semantic Classes.
   > Double_Click to list all the Head Words of that domain.

2. Display all Head Words :
   > Select the option of "List all Head Word"
   > Click twice (Double_Click).

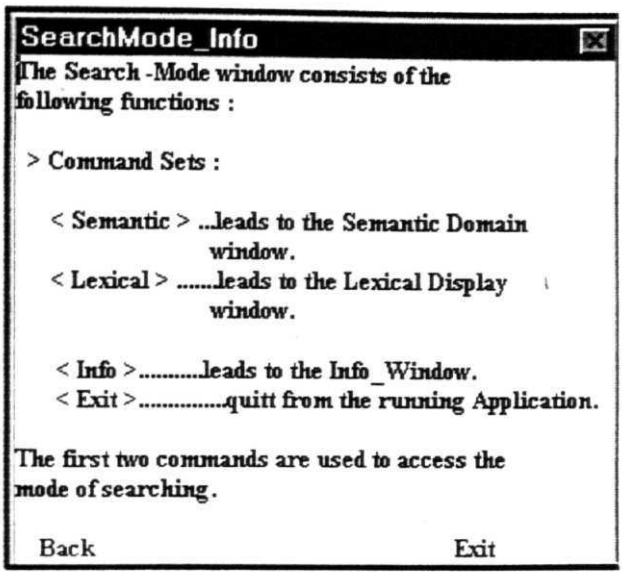
3. Semantic Classes :
   > It contains the semantic classes of selected Semantic Domain.
   > The Semantic Classes will be in alphabetic order.
   > Select and click twice (Db_Click) one of them in order to list its Head Words.

4. Word-List :
   > It displays all the Headwords of the selected Semantic Domain/Semantic Class.
   > Words will be in alphabetic order.
   > If you want to see the details of a word, click twice (Db_Click) on the selected word which lead to the `Lexical Display' screen and appears as the input of `Headword' box.
5. Command-Details :
   > `Back' will lead you to the START screen.
   > `Reset' will clear the screen (i.e. all boxes) to reaccess.
   > `Info' will display the Info-Window.
   > `Lexical' will leads you to the Lexical Display screen.
   > `Exit' will quit the screen.

Buttons	**Applications**
Back	* Go to the Info_window form.
Exit	* Close the running Application.

4.2.2.14. SEARCH-MODEINFO FORM : FORM 7 : SearchMode Info

It includes information about the Command sets used in the Search-Mode form. Its screen display, information, and commands are given below :



Screen Display : SearchModelnfo Form

The Search -Mode window consists of the following functions :

> Command Sets :

1. < Semantic > ..... leads to the Semantic Domain window.
   < Lexical > ..... leads to the Lexical Display window.

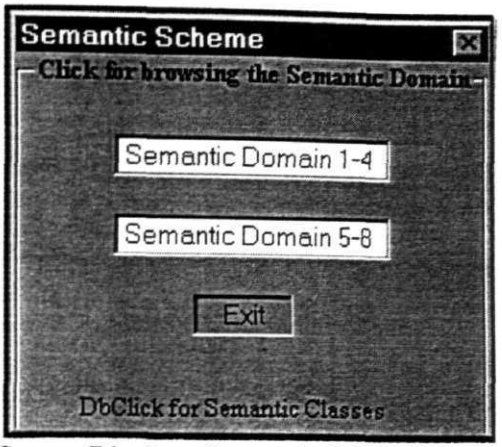
2. < Info >..... leads to the Info_Window.
   < Exit >.....quits from the current window.

The first set of commands are used to access the mode of searching.

Buttons	Applications
Back	* Go to the Info_window form.
Exit	* Close the running Application.

4.2.2.15. SEMANTIC SCHEME FORM

This form consists of the commands and instructions regarding procedures of clicking the box button. The screen display and its commands are shown below :

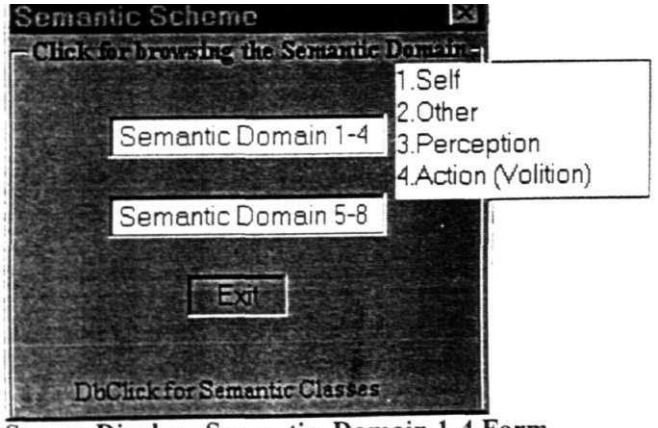


Screen Display : Semantic Scheme Form

Buttons	Applications
Semantic Domain 1-4	* Display the Semantic Domain 1-4 : Classes.
Semantic Domain 5-8	* Display the Semantic Domain 5-8 : Classes
Exit	* Close the running Application

4.2.2.16. SEMANTIC DOMAIN 1-4 FORM

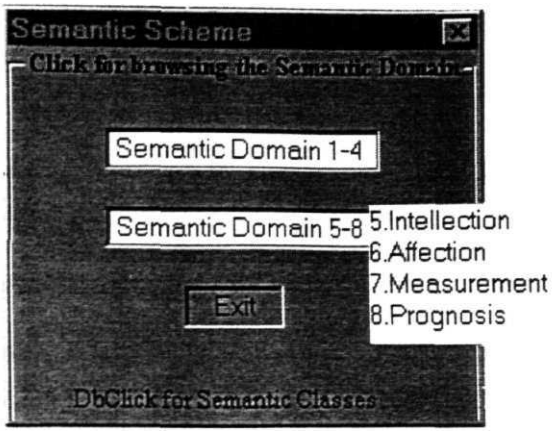
This form consists of the name of the first four semantic domains. They are Self, Other, Perception, and Action (Volition). The relevant screen display is shown below :



Screen Display :Semantic Domain 1-4 Form

4.2.2.17. SEMANTIC DOMAIN 5-8 FORM

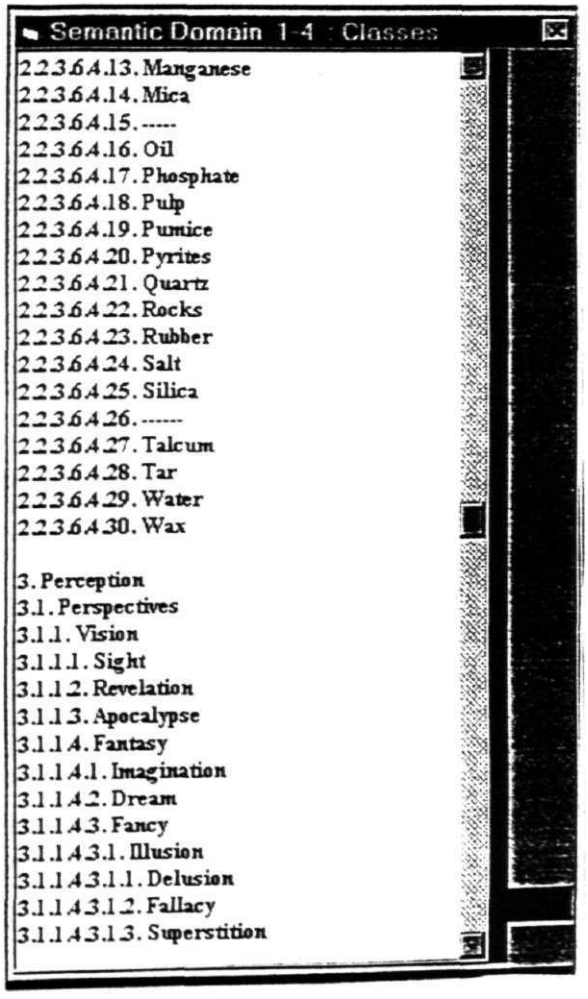
This is an extension of the previous form (i.e. Semantic Domain 1-4 Form). It consists of the other four names of semantic domains. They are Intellection, Affection. Measurement, and Prognosis. Its screen display is shown below :



Screen Display :Semantic Domain 5-8 Form

4.2.2.18. SEMANTIC DOMAIN 1-4 : CLASSES FORM

This form consists of the a list of semantic classes of the first four semantic Domains: Self, Other, Perception, and Action (Volition). Its screen display and the command (i.e. Back) are shown below:

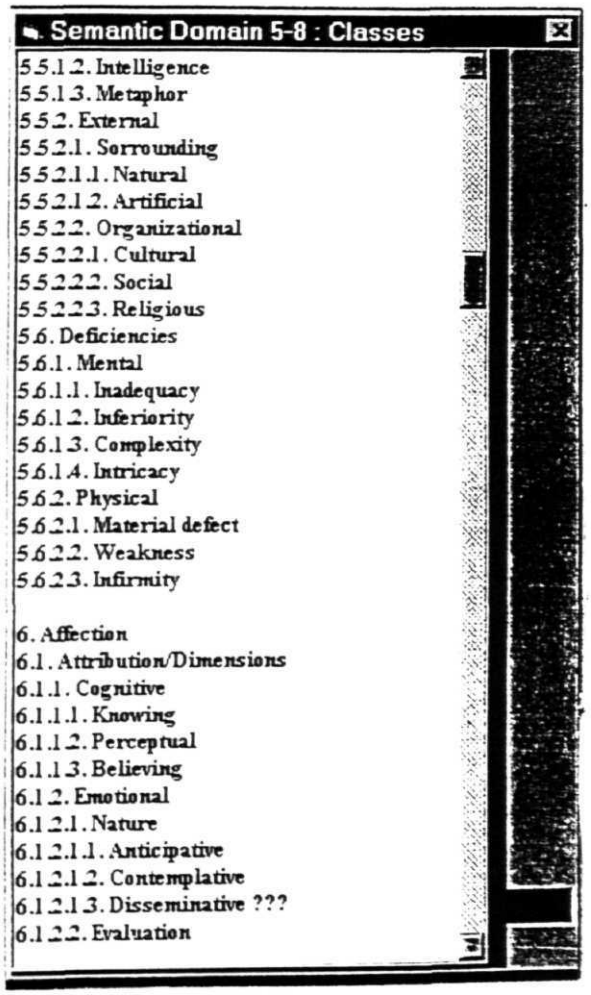


Screen Display : Semantic Domains 1-4 : Classes Form

Button	Applications
Back	* Close the current form. * Opent the Semantic Domain 5-8

4.2.2.19. SEMANTIC DOMAIN 5-8 : CLASSES FORM

This set of Semantic Domains 5-8 embodies a list of semantic classes of the last four semantic domains, namely, Intellection, Affection, Measurement, and Prognosis. Its screen display alongwith the associated commands (i.e. Back) are shown below •



Screen Display :Semantic Domain 5-8 : Classes Form

Button	Application
Back	* Close the current form. * Opent the Semantic Domain 1-4

43. DATA ORIENTATION

The schema followed in the arrangement of the lexical is structured in the pattern of subject domain system dealing with the conception of subject groups with the semantic applicational components. Obviously, the ancient Indian of lexicographical work underlies in the thematic representation and arrangement of entries here. The present module is also presented in the same track of presentation with some fundamental requirements as discussed in the the chapters 2,3 and 4. In this light, the data base structure has been carefully built in order to provide the maximum query facilitation to the user. The development of the required set of data structure depends upon the logical and relational representation of the entry format. As a pre-requisite, the format-mode of the lexical representation, which is mapped by efficacy parameter at the on-line run of the package, is accessed by the standard programming formalism.

4.3.1. DATA : *STORAGE, PROCESS AND ACCESS*

The data storage, as a part of Data Management System (DMS), incorporates its record in a specific format designed as per one's requirement. In fact, the data storage houses all raw materials and functions as an input system to the computer. The data storage is normally governed by the nature of software and its memory location. The system followed in the data storage is undertaken in the form of specified 'fields' which helps one to access it suitably for expected concrete output. The data stored in the machine as an input may have either of the following shapes : character-based, numerical, and sign-based. The data processing is a device used to access data for manipulating a perfect output. The operation can be carried out at many levels :

a. arithmatic operation, dealing with coding system which is retained in the computer memory.
b. operation and re-operation of any type of data which makes it is flexible,
c. sifting, copying , deleting of information stored as part of data can be dispensed with willingly,
d. indexing of particular field is also operated, and
e. an editorial accomplishment - modification and rewriting is also performed.

Recording is an input side of DMS aligning the whole encoding information fed and loaded in the system which can be classified in terms of data-information and programming. To make the computer understand a given task the operator gives the data-information, and to access and manipulate such information he gives some encoding signs considered as programme-recording. Though the term 'recording' in computer is used to refer to a very specific entity related to data feeding only in data-base-file (x.mdb). But here it is used to accredit the whole encoding information of system.

The computing as a process dealing with computation of encoding information as data-material. It can be classified into two categories : numerical and symbolic computation. To keep interaction between the machine-language (artificial language) and human-language (natural language), the user encodes some signs which may be numerical and/or character-based. The computer decodes those signs and tries to comprehend the given information which ultimately results as an output. Besides, the task of computer is to store all information and display on requirement. It provides information in various manner as the user wants to take out.

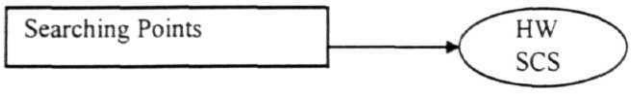
The DMS has an utilitarian process, namely 'sort' which the user can avail of to make the data-material su.table for computation and process with its up-dating particulars The sorting is carried out alphabetically which helps machine to find out the particular information (i e lexical item here) easily. The alphabetization custom can be done according to the language-alphabet-order wh.ch requires one to give the information regarding computer. The sorting and resorting of the same or even the other type is possible. The information loaded in the database file (x.mdb), having been sorted out, can be used to search the particular given information The searching is only conceivable when the loaded data has been alphabetized. The utilization of this process to know about the particular information may be done either at the macro-level - displaying all information, or at the micro-level collecting only restricted information as part of 'macro-information'.

The ultimate aim of a user feeding the data into the system as he/she uses it is to acquaint him/her with the primary objectives. The process of displaying can be carried out on a two-folded observance : the monitor-screen during the data feeding (input-time) or even after completion to view the partly or fully fed data-material and as final output passing through the system to printer for printing display. But even in the case of the second, one can display only through a part of the whole, as the 'display' is a visual process. The purpose of developing any software is to provide more facility to the user. To fulfil this goal, the computer scientist has to prepare a software which can be copied and distributed among others in order to convenience the maximize utilization of it which would be beneficial to all. The 'copy' of encoded data can be copied from the hard-disk to diskettes/floppies or vice-versa. The user can take such copies from one terminal to another. The "installation' of software is also one kind of copying process which is related to the copy of •command.com' file and other affiliated ones from diskettes to hard-disk.

4J.2. DATA STRUCTURE -.*ATTRIBUTIVE NETWORK*

The data structure followed in building the present electronic thesaurus comprises the prototype conceptual attributes. The attributive network is framed out and co-related in the following diagram which is based on the algorithms:

Algorithms

- Spelling variations (sv) and Identical Word (iw) are equivalent to Head Word (HW).
- 

The attributes of the entry are given in the following diagram along with their interrelationship which is shown by the arrows. The retrieval of the data-application is also marked within the square box.

4.3.3. DATA- STRUCTURE-DELINEATE

The interrelationship among the fields is shown in the following diagram The inter-network of fields can be represented as follows :

1. Condition :

- Enter the headword in Headword Editbox of the Lexical Display Form or select from the Listbox of the Semantic Domain Form.

2. Process :

- It will search the headword in "entry" table. If found = True Then
- All 15 attributes, viz. grammatical class, semantic domain, semantic class, synonyms, spelling variation, identical word, related word, antonym, derivational word, compound, collocation, root definition, illustration, and example in their respective boxes.

In order to retrieve all these information, the following type of database has been prepared. Here, we see the logical interlink among the fields.

- The headword is the main point of search.
- There are four fields for four grammatical classes, viz. Noun, Verb, Adjective, and Adverb, respectively.
- Each grammatical field is related with three semantic domain fields, therefore, there are twelve semantic domain fields. Their distribution is given as follows:

$GC1_{(Noun)} = SD_{(1\_3)}$
$GC2_{(Verb)} = SD_{(4\_6)}$
$GC3_{(Adjective)} = SD_{(7\_9)}$
$GC4_{(Adverb)} = SD_{(10\text{-}12}$

- One semantic field is related with one semantic class field, therefore, there are twelve fields for semantic class, too.

- Each field of semantic class is related to four fields of synonyms. Thus, these fields are distributed as follows:

$SC_{(1)} = SN_{(1\_4)}$
$SC_{(2)} = SN_{(5\_8)}$
$SC_{(3)} = SN_{(9\text{-}12)}$
$SC_{(4)} = SN_{(13\_16)}$
$SC_{(5)} = SN_{(17\_20)}$
$SC_{(6)} = SN_{(21\_24)}$
$SC_{(7)} = SN_{(15\_18}$
$SC_{(8)} = SN_{(29\_32)}$
$SC_{(9)} = SN_{(33\_36)}$
$SC_{(10)} = SN_{(37\_40)}$
$SC_{(11)} = SN_{(41\_44)}$
$SC_{(12)} = SN_{(45\_48)}$

- Other attributes are directly related to the headword. Their number of fields are given in the Table under 4.3.5.1.

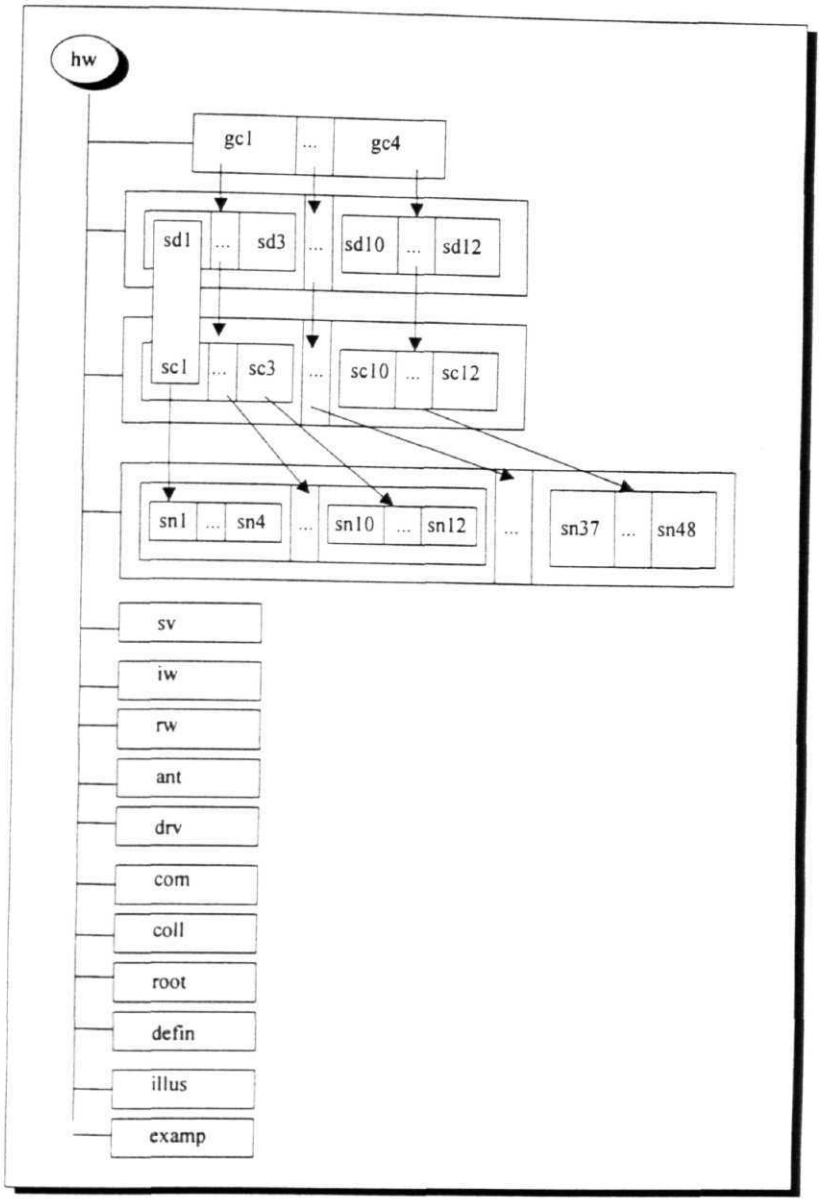
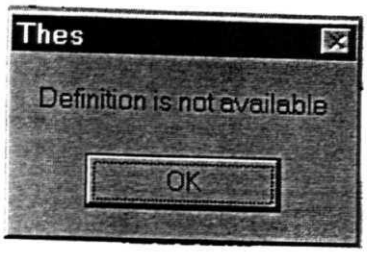


Diagram 12 : Showing Data Design and Interlink among Fields.

4J.4. VB : *DATA ACCESSING*

Visual Basic is facilitated with the Data Manager which is activated by using the Window command from the File Menu. This Data Manager supports the database to the Visual Basic application by accessing the data. The data accessing deals with the manipulation of data stored in the database by managing its objects. Here, we have selected some screen displays from the Visual Basic to show the process of creating the database step by step which enables us to understand the whole process of data accessing easily. If information of a word has not been fed in the data file, the message-box will appear with the message - "The (searching information) is not available". For example, if the definition of a searching word which has not been fed in the datafile, the message - "Definition is not available" will appear on the screen as shown below:



Screen Display : Message Box.

The Data Manager supports the Microsoft Access, FoxPro 2.0, FoxPro 2.5, dBase III, dBase IV Paradox 3X and BTRIEVE. Normally they can be divided into three catagories : Native Databases, External Databases, and ODBC (Open DataBase Connectivity) Databases. The database types and their supporting versions are followed by the the Visual Data Manager screen display. The Data Manager can be open in the Visual Basic in the following manner:

4.3.4.1. OPENING DATABASE

For opening database on MS Access version 2.0. the following steps are required to follow :

- Go to the Add-Ins Menu and select the Visual Data Manager command and click it.
- As soon you get the Visual Data Manager screen, select the File menu. There are two options : In the case of new databases, select the New Database command from the File Menu from the Visual Data Manager screen and then select either Access 2.0, or in the case of an existing database, select the Open Database command from the Data Manager, as shown above, and then select the supporting database and click it.
- In the case of open database, you have to select your .mdb file from the Open Database window and more forward as shown below.
- As a next step, interact with the Database window where you can select New command for opening a new table and Open command for the existed table. The Database window-could then be seen.
- It may be noted here that in the case of existing table, the Open command leads you to the so called the Editing Record screen (it is not an exact name). But in the case of New command, it leads to the Create New Table window.
- After writing the name in the Table name box, press the OK command button which leads to the Editing Record screen which is an environment where the data can be entered and stored.
- Before going to the Editing Record screen, one has to go to the Table Window to create the fields and indexes.

* r" ZA Cnter thC fiɛld namC in the Table Window, type the field name in the Add Field Window.

- In order to make the Index file, enter the information in the following window, i.e Add Index Window.

4.3.5. THES.MAK PROJECT : *DATA DESIGN*

4.3.5.1. DATA TABLES

In this module we are using following three database tables. The structure of each database table is given below. The common field in all three tables is "hw" i.e. Headword. Through this "hw" we can only extract the other information from each table. We could now display each database in detail.

- Each small square bracket here represents the name of the field of the corresponding database file.
- The dots represent similar types of fields.
- The arrow line indicates the relation between two files.

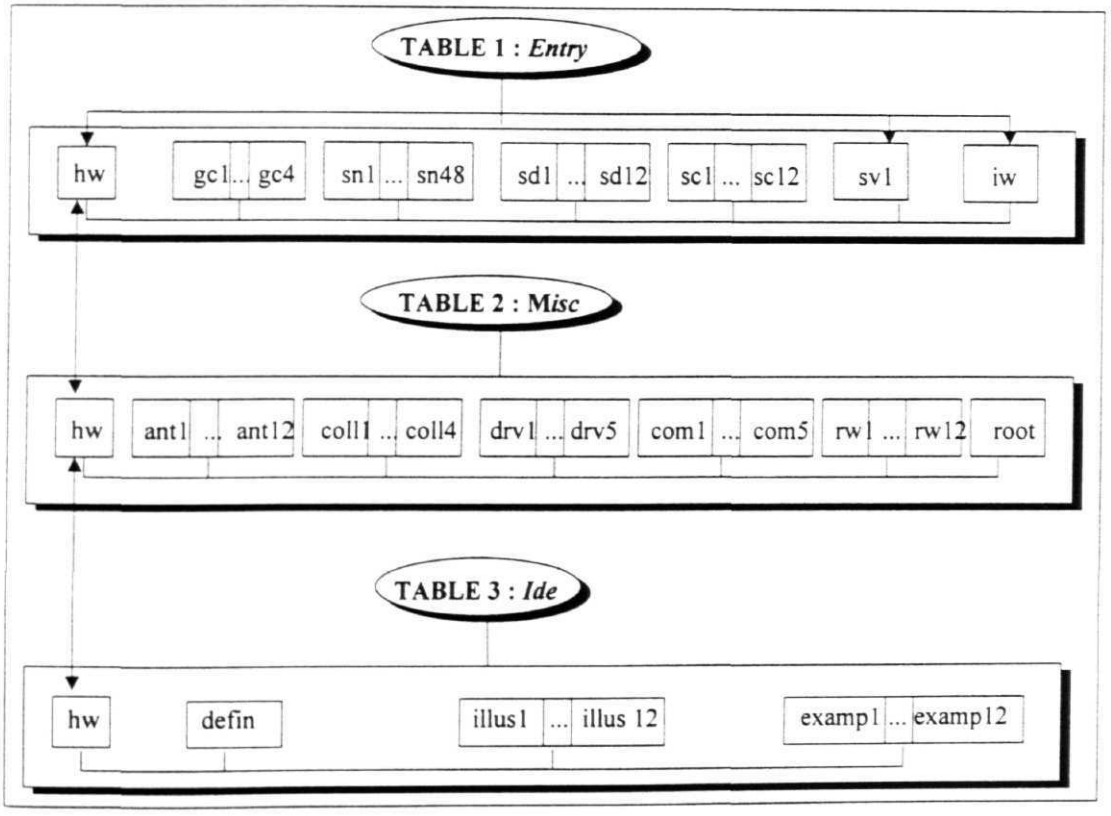


Diagram 13 : Three Database Tables.

4.3.5.2. DATABASE : *FIELDS AND WIDTH*

The following box showing the "ABBREVIATION OF *DATABASE-FIELDS"* will enable the user to identify the terms given in abbreviation.

hw .....Head Word
gc .....Grammatical Category
sn .....Synonym
sd .....Semantic Domain
sc .....Semantic Class
anto.....Antonym
collo.....Collocation
derv .....Derivative
com.....Compounding
iw .....Identical Word
rw .....Related Word
sv .....Spelling Variation
defin.....Definition
illus .....Illustration
examp .....Example

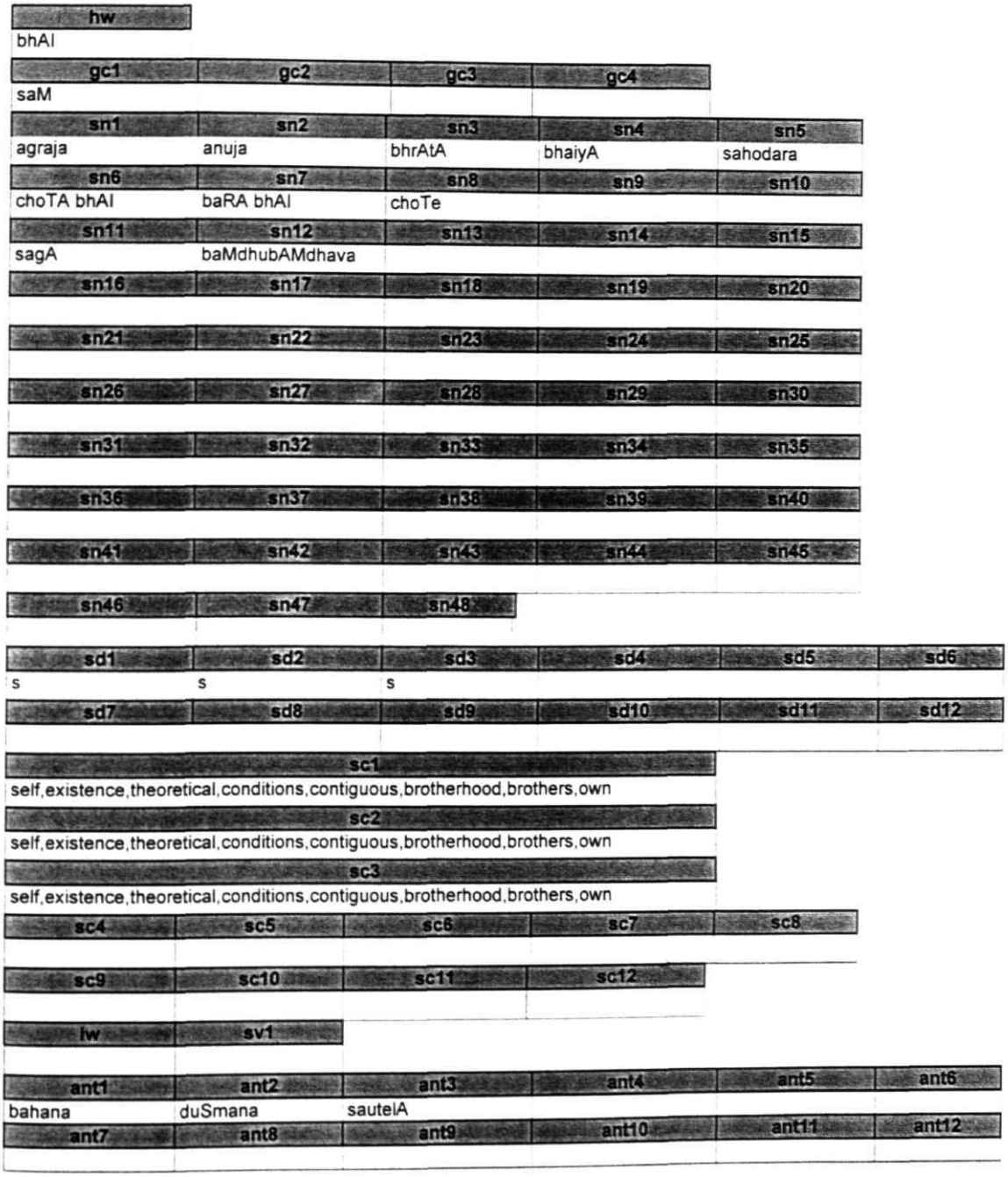
The structure of database files is as shown in the box given below. The first column is the "Name of the Field", the second column is the "Width" , the third column is total no. *of* fields and the fourth column is the "the total width for a particular field in each file:

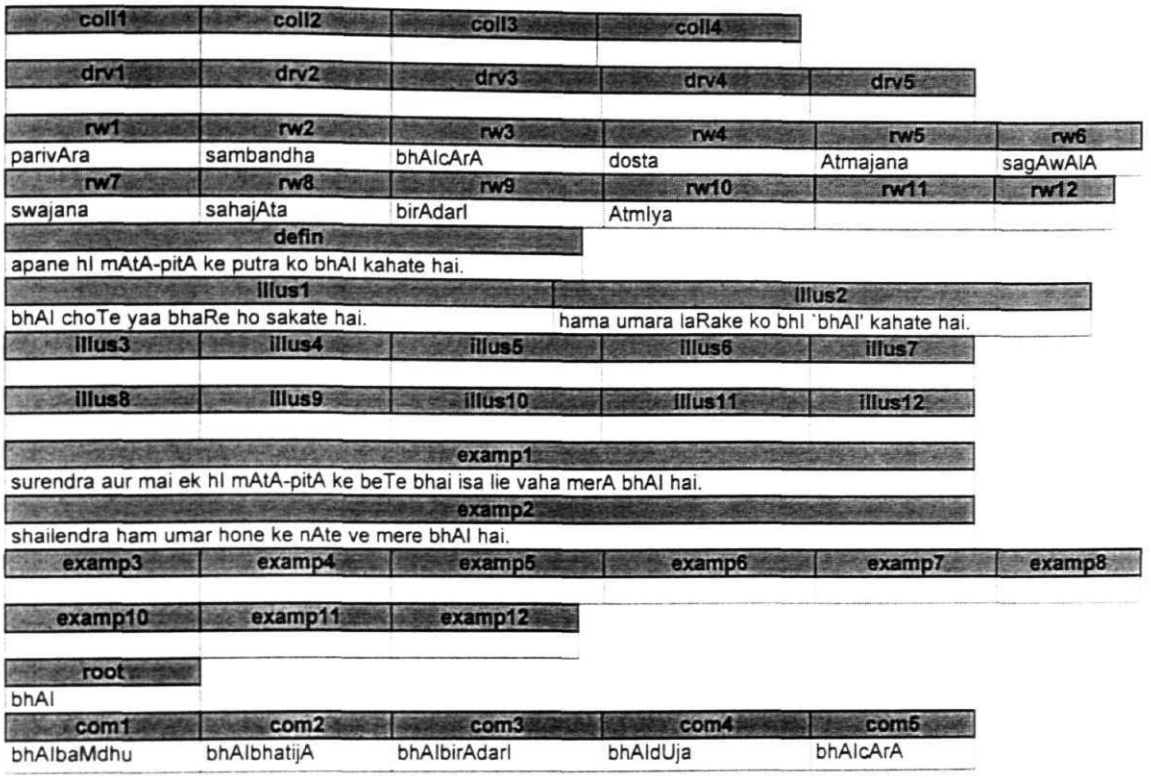
Abb.	Width	No. of fields	Total Width
hw	20	1	20
gc	10	4	40
sn	20	48	960
sd	3	12	36
sc	150	12	1800
sv	20	1	20
iw	20	1	20
rw	20	12	240
ant	20	12	240
drv	20	5	100
com	30	5	150
coll	30	4	120
root	20	1	20
defin	200	1	200
illus	150	12	1800
examp	150	12	1800

Total (field-Types) ⟶ 16
Total (fields) ⟶ 143
Total (width) ⟶ 7566

4.3.5.3. EXAMPLE

There are , in total, 1291 records in the present thesaurus-module. Thus, 1291 entries are created and appropriately filled in the present database. Here, it may be noticed that all information about each entry has not been enterd into the database. The aim of this database creation is to show how the system has been designed and the mode of access, rather than making it an absolutely exhaustive database. Here, the Hindi word bhAI 'brother' which has benn discussed earlier could be selected once again to show how the information about this word has been allocated into the database:





4.4. SUMMARY

This application is developed in Visual Environment and all the advanced features are used to make this product user-friendly. The bi-directional access, i.e. from lexical item to semantic class and from semantic class to lexical word, is the speciality of this product. For storing the data we have used MS Access 2.0. Both are Microsoft products, and, therefore, it easy for data handling. While designing the application we consider all the cases related to the GUI application. On-line information about the package application is available.

CHAPTER 5

CONCLUSIONS

"The lexicographer's situation is frequently comparable to that of the man who tries to get at the meaning of the traffic signs while driving rather than to that of the man who has a perfect knowledge of the abstract system before he begins to drive. "

-Zgusta(1971:26).

5.1. THESAURUS AS A COMPLEX WHOLE

5.1.1. CLASSIFICATION

Any 'thesaurus', as the name indicates, is a treasure trove. A thesaurus as an entity has several layers of distinction - (i) *general* - dealing with all human knowledge or *specific* - dealing with a subject matter limited to a selected area of human knowledge or application, (ii) *monolingual* - written in one language only, *bi-/multilingual* compiled in two or more languages, or (iii) *manual* - prepared for manual reference (and often prepared manually), and *electronic* - accessed through the computer.

The proposed electronic Hindi thesaurus is a monolingual database aimed to act as a wordfinder based on the semantic domains and is also designed to act as a synonyms treasure-trove. It is also a mapping system which is dynamic due to the lexical functionality approach followed here which helps the lexicographer to solve theoretical as well as practical problems he faces while compiling a lexicon. There are many parameters to access the model undergoing certain lexicographic network in order to map the lexical unit and its associative components. The complexity of network framed with its own enhancement depends upon the range and space of semantic shades of a lexical unit, reflecting through the semantic domains. This idea can be conveyed through the following diagram :

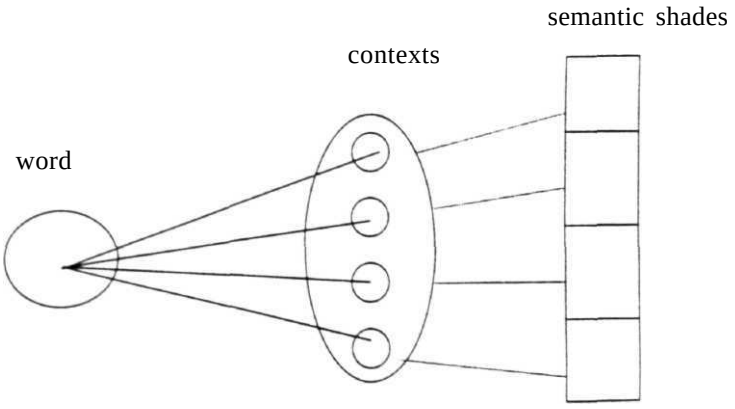


Diagram 14 : Complexity of the Network.

5.1.2. RECONSTRUCTION OF SEMANTIC SHADES

To understand the semantic shades, one requires world knowledge - something which all human beings have, to different extents, rather than having a specialized knowledge of certain disciplines (or its allied subjects). The meaning of a word is itself a product of combination of all related disciplines at the sametime, derivable as a part of world knowledge. Due to the limited human capacity and capability of acquiring and storing knowledge, one usually perceives (and 'knows' of) only certain shades controlled by or related to bound field(s). The reconstruction of all semantic shades of an entity is a collective phenomenon for which one has to depend on the whole speech community or even sometimes go across the community (and consider the genetically and/or culturally related languages). When a speaker expresses something, there are some intentional impact found in his speech that pragmatises his statement. On the other hand, the hearers and the readers are guided by their own pragmatics. There may be differences between these two different sets of pragmatics for various reasons - which is what leads to additional reading or interpretation to misreading. This pragmatics becomes an essential aspect of the dictionary or thesaurus. This is why, Zgusta (1971:24) rightly points out, "The lexicographer's activity, however, can be conceived as rather pragmatic one, at least in this respect." He is absolutely correct in his statement because the intention of speaker is an essential element in the expression which is governed by the situation.

5.1.3. WORD EXTRACTION AND SENSE DETERMINATION

A lexical item is nested within a number grammatical categories and the senses assigned by it depend on the categories as well as on the given situation. Indeed, the lexicographer's approach towards language analysis is to explore its general usage and identify its potentially multiple grammatical functions and place them in intertwined linguistic components. The construction and application of lexical units often display common and/or partially overlapping features which enable the lexicographer to compare the systems that lie behind any two given words. In terms of features two or more lexical items can be identical, but at the macro level of application they differ from one another (howsoever slight difference that may be). This type of dissimilarity or contrastive feature-combinatrons can be noticed at a glance which get flashed out on the screen showing coherent semantic applications specific to an individual word or word application. Hence. the task of an electronic lexicographer is beyond the normal lexicographic norms but he must be a semanticist or a lexicologist. Of course, he must also be a man who specializes in many other related fields - like a conventional lexicographer - in fields such as morphology, phonology, orthography, etymology as well as in different literary genres. He must also examine all words critically and in terms of their applicability, rather than only in respect of acceptability, of the lexical function as constituting the language system. The senses can be determined when the extraction occurs at the lexical, sentential, or discourse level. The sense of a lexical item is basically extracted from certain definite spheres, whereas while occurring it assigns this sense in combination with other senses to complete the meaning of a clause or a sentence. -Recognition" of such processes may be difficult in the massive data taken from the field in the case of a lexicographical field work and even in a composite document on which the lexicographer must work to find out the complete and exact sense of a particular lexical unit and then set it up into a its proper locus. The degree of 'Complication' of this extraction process depends upon the degree of difficulty of the text and on the ambiguity which may refer to the contexts from where the lexical unit is extracted.

5.2. COMPILATION FEATURES

5.2.1. CLASSIFICATION OF KNOWLEDGE

Though in the compilation of a thesaurus, a lexicographer analyses the extracted inflected and contextually alive word as lexical units and incorporates them as a part of his practical system by adopting some rules and method, the semantic shades of a lexical items can be treated as **equivalents** in the case of a bilingual product and as synonyms in a monolingual product. The analysis is carried out according to the nature and type of thesaurus either in the synonymous or in the semantic mode. It has been shown, as a part of the present research, that the infrastructure necessary in this system of compilation is a massive subjectwise classification of knowledge. The entirity of human knowledge has been tried to be classified here in terms of certain broad and specific classes in a hierarchical framework. No doubt such practice goes back to the ancient tradition in the Indian lexicographic works as we have seen in the second chapter with some sample of entry set up in Indian classical language, i.e. Sanskrit. However, the whole schema is made on the assumption that the user will be facilitated with this set of semantic classes under which specific lexical items occur, (possibly in more than one slot), and that it will be a more comprehensive tool than a traditional dictionary or even a monolingual manual thesaurus.

5.2.2. DESCRIPTION ORIENTATION

The *description orientation* is an important factor of compilation. It can be noticed that enhancement of description in the dictionary requires a sufficient amount of space where the lexicographer can give precise references about each individual entry. But, this is not open-ended. The lexicographer has to limit himself somewhere. But the question is where exactly the restriction should be imposed? The answer lies in adopting a two-way solution : firstly, leaving one or more individual component(s) out of the total references, by highlighting on certain features, and secondly, in selecting the total component by removing the details about them. The question of arrangement of the various usages of a lexical entry or sub-entry under a particular entry is itself a fundamental clue to the system formation. Arrangement of entries could vary from dictionary to dictionary. In this regard. Kelkar (1973:62) says. "... in a descriptive, synchronic dictionary the arrangement should be in an order of descending frequency as seen in the attestations of the period under consideration; in a historical, diachronic dictionary the arrangement should be in an order of increasing frequency of the first attention of that sense."

5.2 J. SPECIFICATION OF PURPOSE

The *specification of purpose* varies depending on the kind of reference or help one expects from the dictionary. One can, of course, ask a question about the "expectation of reference' i.e. whether **the** expectation can be marked out and, if so, whether the reference is provided by the system (or the product) as per requirement. But still, there are some kinds of overlapping which are inherent in the purpose just like the overlapping of lexical units in compilation. Because of the specification of purpose, different types of lexicons come into practice by following different models. Gajendragadkar 1973:13) gives his opinion regarding the above statement by stating, "There are different types of Dictionaries satisfying specific needs of various types of seekers of knowledge. Apart from the specialised ones like the Dictionary of scientific terms or philosophical terms which are intended to cater to a selective group of people, looking for specific information."

5.2.4. EVALUATION

The motivation of *evaluation* towards the sense appreciation is factually dependent as one can see in Kelkar's (1973:63) statement: "The critical test is whether the meaning is the one which will most readily suggest itself to the user if the vocable is *mentioned out* of the context rather than *used* within a context. In relation to a synchronic account of a language that is no longer current, one may want to recast this test slightly : the basic meaning is the one that an editor interpreting and annotating a text is most disposed to pick up so long as there are no clear contextual pointers in a different direction. In the rare cases where a vocable may have more than one equally viable neutral senses." Moreover, the evaluation can be carried over on the acceptability and utilisation of the lexicon. The dictionary has to meet both these criteria adequately, as has been taken care of the present compilation. For instance, the semantic thesaurus should provide the human knowledge as well as the synonyms under their semantic classes. Likewise, the data collection as suggested here is also well justified. The lexicographer has to examine the source(s) for his compilation. For instance. the sources for the historical use are ancient and medieval literature and the existing dictionaries of an earlier period. The sources for present use are obviously open-ended where both written material and oral speech (recorded/noted/observed by the compiler) are important. The treatment of polysymy and homonymy are also defined properly and represented in the appropriate locus. It is also admitted that grammatical and morphological information will be tackled in the required fashion. The representation of synonyms (and equivalents in a bilingual product) should be mapped at the scale of 'closest to closer". The closest synonyms should be given priority over the other.

5.2.5. **THE** ARRANGEMENT

The nature of thesaurus determines its comparability. A thesaurus concerned with form and meaning of words and their correct usage has scope to give scientifically accurate details about the meanings of words. A thesaurus is meant for giving the contextual information to users. Moreover. it should also provide a brief account of the semantic significance when a word occurs in a particular context. For instance, in Amarkosha (cf. Chapter 2), Amar Singh arranged the entries under their semantic shades/domains. In thesaurus, meaning becomes the heavier among the inbuilt systems. The metaphoric senses are also concerned consciously alongwith the contextual account. The lexicographer should be aware of the possibility of the meaning having some additional functions either in plain speech or text or in metaphoric expressions. The context should, therefore, be fully reflected in the way a thesaurus is constructed, even if it is not directly quoted overtly as done in the dictionaries. As one can see, this requirement has also been taken care of in this case.

5.2.6. **THE ECONOMY** MEASURE

The lexicographer, in compilation of semantic thesaurus, follows the tactic *of economy* in both cases : (a) Entry Selection and Representation and (b) Semantic Determination. It is obvious that one should not generally select a monosemantic lexical item because such items are best described and tackled in dictionaries. But anyway, in the presentation of polysemy this "economy" may prove to be a fruitful methodological principle in a semantic thesaurus. The assumption is that multiple meanings should be presented in an economised manner rather than giving a descriptive information. The maximum range of reference should be phrases consisting of sometimes two or sometimes three words. There is no such methodology of the initial meaning or even that of the secondary meaning in an electronic thesaurus. First, one can search each item individually. If one

searches the synonyms at one glance of a single screen display, he will follow the initial reference in the first place, and then the secondary ones by adopting the procedure of 'closest to closer'. The following situat.on may be possible when meanings undergo initial and secondary assessments. Recall the statement of Mehendale (1973:77) who advocates Thieme's view regarding determination of initial meaning by stating, "This is an excellent methodological rule. Of course, it cannot, by its very nature, be applied to words occurring only once, or even twice, but in cases where a given word occurs more than once or twice, and if we are fortunate enough to notice that the contexts in which it occurs are *different*, the above method deserves to be tried out; not only this, it may be asserted that it is the only proper method to be followed. And if in its application we are able to come out unscathed with only *one single* word with one single meaning, in spite of the fact that the contexts are *different*, we are very likely to be nearer the truth."

5.2.7. THE DEGREE OF HOMOGENEITY

Thesaurus aims at dealing with the vocabulary of a language from an evaluative aspects dealing with all semantic range of the lexical items of the language from the possible available data. Hindi thesaurus would also need to represent formal selection of spelling variation. It is noticed that like other features, there is spelling variation found from region to region of the same speech community. Such phenomenon marks on the degree of *homogeneity* - complete or partial. In the case of complete homogeneity, the nature of language is the same whereas in case of partial one, the nature varies from region to region of the same speech community. Therefore, there is a variation lying in the language treatment as dialectal. Though the dialectal or ideolect is not our concern in the present research. But, we do have a glance of these because they directly or sometimes indirectly affect the semantic thesaurus. In other words, they can make the classification much more complicated. But there is no way to escape from such facts. They may affect in the entry selection and determination *of* the semantic assignments of the lexical items; where the semantic shade of even individual lexical items become vast. The most important aspect of human knowledge is that of understanding of all possible orthographic as well as semantic variations occurring among different dialects of the same language, which should all be noted in the compilation. If this is done, it will facilitate the user to not only for the lexicographic reference but also dialectal reference. Because of limitation of scope, this however could not be done in the present prototype electronic thesaurus to the extent desirable.

5.2.8. SENSE ASSIGNMENT : *THE SEMANTIC AND THE PRAGMATIC*

To understand the pragmatic aspect of the speech, especially of the lexical unit, one has to know that the *concretization* of semantic assignment may vary from one speaker to another because of regional differences. Moreover, the pragmatization of speech is also dominated by the speaker's mode of expression. Therefore, *the feeling* comes into the centre of speech. Such evocativeness is an individual phenomenon of communication. But, due to the certainty in the actual speech or parole, the hearer can recognise the real intention of the speaker. The speech is affected by the speaker's sensitivity, socio-cultural background, education, and the personal touch of the surrounding.

The sense assignment may vary because of the variations of different kinds in the language. A knowledge of these variations will help lexicographers to determine the primary sense and secondary senses. It will also enable him to compare the sense ranges at the axis of time, space or other variables. Through this faction, he can judge the future utility of his thesaurus. When he plans

to compile a thesaurus, he must have a clue of the periodic or the geographic span he is going to cover in his thesaurus. The *ideology* is another concept in the planning process which determine the locus of the entry set-up. As a part of systematic representation, it encloses the rationality of semantic propert.es explo.ted as complete or partial synonymy, although it is more appropriate in preparing of a synonym dictionary, i.e. alphabetic thesaurus, where the lexical units are selected and organised in systematic framework.

5.2.9. SENSE-TYPES

The interconnection between two senses in lexical combinations may be of three types : First, two senses are completely interrelated to each other in the maximum contexts. It will be logically and philosophically wrong to say that they are identical in all contexts rather than claiming they overlap in the "maximum' number of contexts. Secondly, there could be partial overlaps or interrelationships between two senses. It means they are largely identical and to restricted certain situations. Thirdly, when two senses are not identical in any context, they just function as dissimilar or unidentical, but may or may not be actually antonyms. The task of a lexicographer is to analyse language and point out the nature of interconnection right at the beginning of one's work. Logically, it is impossible (obviously not in all cases, but in certain cases) to indicate the range of lexicographic context. Normally, this is so when the expression is either unclear in recognition or having clumsy range of assignment. The immediate context also fails to give the final clue or the exact picture of sense assignment. In the study of language in which a thesaurus is prepared, it may happen that sometimes a clear cut context is preserved which makes the identification of lexical meaning recognised. The nature of thesaurus makes the lexicographer keep different attitudes towards the sense assignment occurring in the specific contexts. But the lexicographer is generally advised to cover almost all possibilities when he prepares the thesaurus.

5.2.10. THE IMPLEMENTATION

The effective implementation of all aspects of sense assignment, their interrelationship, principles, and processes discussed above depends upon the knowledge of the lexicographer. If he fails to recognise even a part of the whole network, the compilation would not be successfully carried out or it may be that the aims and objectives of compilation would not be met. Here, regarding to the ability of lexicographer, one must emphasize that the decision taken by him must be nearly perfect on the account of purposes. Recall Hayakawa (1941:55) statement in this context. "The writing of a dictionary is therefore not a task of setting up authoritative statements about the true meanings of words , but a task of recording, to the best of one's ability what various words have meant to authors in the distant or immediate past. The writer of dictionary is a historian not a law giver." The task of compilation requires no small delicacy and knowledge. But it requires the lexicographer's vital knowledge of language. The lexicographer who compiles a monolingual dictionary should be a native speaker or native-like speaker of the language.

5.3. COMPUTATIONAL PROFILE OF THE ELECTRONIC THESAURUS

5.3.1. THE FLEXIBILITY

An initial look at the possible counter-example of the product-prototype developed here will provide an overview of flexibility of computational lexicography, especially in design and scope of it, will make us realize that "Computational linguists and large industry developers recognise that, for real world application, it is of fundamental importance that natural language processing systems are able to deal with tens and even hundreds of thousands of lexical items." Hence, one must achieve a reasonable degree of progress in the lexicographic work in the NLP tradition which will enable the computational linguists and computer scientists to achieve the ideal goal of "automation" in language processing. Preparation' itself is a knowledge-based task depending upon the nature of its utilisation. In such respect, the lexicographic work, in general, comes under the semantico-pragmatic knowledge where the user should be able to identify the perspectives of lexical entry in lexicon. The dimension of preparation and utilisation of an electronic lexicon is due to the nature of work, whether it is machine-readable-dictionary (MRD) or manually printed dictionary (MPD), and the information to be included will enable us to utilise it potentially to any NLP system. Broadly, an exhaustive lexicon takes into the detail morphological and grammatical information on one hand and semantico-pragmatic information on the other. Yet relevant variants related to pronunciation and graphic formation, including hyphenation, and problems emanating from them must also be possible to tackle by now. The present lexicon is to be viewed only as a step towards that direction.

5.3.2. THE APPLICATION

As far as application of computer is concerned (in the compilation of lexicon), especially the semantic thesaurus, the lexicographer has to update himself as a computational lexicographer in order to recognise the application and utilisation of the developed software. At first, he has to learn about the efficient software. The aims should be purpose-oriented rather than for name sake. It requires a critical assessment by keeping the user's requirement in the view. It does not mean that he can restrict himself to the newly defined programme, but if he is flexible enough to select any model or system which satisfies the maximum requirement of the user, the real credit goes to the software which has such efficiency that if the user wants he can perform rather than get tossed down at the programming. It is also true that the efficacy a run-time system has to depend upon the programming utility and algorithm. If the programmer, who may be a computational lexicographer already, selects a high performing algorithm with a high power system, definitely, one can not only access the codes used in developing the software, but can also test the principles followed in building the programme. Once again, the work here with a limited set of 1291 headwords is intended to test out the power of the programme which has been written.

5.3.3. THE RELIABILITY

The reliability in productive power of a programme keeps the chance of generating a good output. There is also a possibility that the programmer fails to meet the defined target, but his attempt and logical solution might help him to out come from such troublesome situation. In such case, he has to take the following steps : recognising the defective points, sort them out by classifying according to their nature, fixing the proper location, and recodifying as need. When the

user wants to change something which according to him in his own view or requirement, the function of the programme should be up-graded. Perfection lies in the coverage of subject matter, flexibility in system utility, specification in programme, and the concreteness in the output.

5.3.4. THE DEGREE OF AUTOMATION

Though the task of a programmer or developer is to produce a useful and an automated system, the *degree of automation* here becomes a more important quality. This is especially so in the case of thesaurus compilation depending upon whether it is developed for the machine or for manual use. In this case, of course, the model that has been developed here is mainly designed and developed for the machine. This thesaurus can be accessed through an on-line system. Here, the components of the proposed thesaurus are well defined and are nested in the system. It is true that to complete the package, there are many system requirements such as the right kind of hardware, or software support for the system, a good compiler and analyser besides efficient human resources as operator, programmer, and designer, and documentation support. This should be eventually made importable into different word processing softwares so that those using any Hindi Word Processor could make it a part of their help kitty. There would, of course, be constraints restrictions of the system requirements.

5 J.5. THE DBMS AND PROGRAMMING IMPLEMENTATION

The DBMS is a very complicated computational system, where database, as an integrated collection of documentation files can be accessed automatically, consisting of data element at a lower level. The whole design in the present database is arranged in a hierarchical pattern. The data file consists of different records and each record has many fields (cf. Chapter 4). The information stored in the field is directly retrieved from the database as an output. To attain such expected targets, the whole process is set up in the mind from input to output in the application of pre-processing and post-processing. Later, this module is set up in the system in order to run the abstract model in an automatic fashion. The data structure and the interrelationship among different components at the different levels are primarily concerned with the data designer. The existence of any software is not possible until it is well implemented. The programming is an outcome *of* mathematical, symbolic, and logical operations where they are internally related. Programming makes it possible for the computer to understand the nature of tasks and execute them. As is well known, there are two types of programming implementation as available in personal computers (PCs) and they are explained below.

DBMS is used as a device in creation of a database system prepared within the constraints of the software to implement the small -.mdb in order to get an output instantaneously. It is a ready-made programme which works to generate a pattern for a given material. Noticeably. such a device can be used for very small processes. But for access to the heavy load data-material. one would have to prepare another programme for implementation. The 'programme' is like the sense of the computer which directs to perform the expected accomplishment. The programme-file is a separate file made of defined characters, and arithmetic codes are represented logically in the **form of matches** and conditions. The programming can be nested by several sets of sub-programming within it due to certain exigencies or requirements. Thus it can be named as as the **main programme** and its sub- programme(s).

The system having been completed as a part of one's software package-building has to pass through the test of capability and accessibility. The first of these tests judges the power of .mplementat.onofag.ven task under a prescribed input in relation with its output, whereas the latter accesses the programme in terms of time, utility, and flexibility. This accessibility accomplishes the instructions given to the computer. The whole issue relating the interaction between man and machine (in this case, the computer) can be summed with the statement of Andres (1980:146) : "... the dictionary-maker will still need to use his intelligence and skill to execute those phases of his task that require more thought. However, once lexicographer's "raw material" - the language data - have been adequately prepared presentation to the computer, and once the machine has been supplied with the instruction on how to do the job, it can do the work much more accurately and thousand times faster than any human being ever could."

The database structure has been precisely built in order to provide the maximum query facilitation to the user. The development of the required set of data structure depends upon the logical and relational representation of the entry format. As a pre-requisite, the format-mode of the lexical representation, which is mapped by efficacy parameter at the on-line run of the package, is accessed by the standard programming formalisms. Identification of database in terms of functions and its performance require items to enable the developer to retrieve the document stored in the data file(s) as and when needed and in the manner needed . The well structured scheme is based on the ability which may be classified as follows :

i. Programmers => Native-like ability : Special knowledge.
ii. System => High run-time technology with high efficacy.

As one could see both types of programme implementation and running through the tests were done in developing the present prototype product.

5.4. FUTURE ORIENTATION

The future of developing full-fledged electronic thesaurus lies in learning from the past. There was a time when the scholars like J. J. Scaliger and A. Gleason were skeptical about dictionary compilation when they gave out their view in the following words (as quoted in Zgusta 1971:15): "The worst criminals should neither be executed nor sentenced to forced labour, but should be condemned to compile dictionaries, because ail the tortures are included in this work." Similarly, Gleason had pointed out, " Dictionary making is tedious in the extreme. It is exacting. It is an incredibly large job."

In contrast to these feelings, let us see what Zgusta (1971:357), had to say. "Lexicography has also its charms and rewards." and to corroborate his own view, he presented the opinion of J.R. Hulbert as an additional justification. J. R. Hulbert had stated : "I know of no more enjoyable intellectual activity than working on a dictionary. Unlike most research, lexicography rarely sends one in fruitless quests; one does not devote days, months, or even years to testing an hypothesis only to decide that it is not tenable, or to attempting to collect evidence to prove a theory only to have to conclude that sufficient facts are no longer in existence to clinch it. It does not make one's life anxious, nor build up hopes only to have them collapse. Every day one is confronted with new problems usually small but absorbingly interesting; at the end of the day one feels healthily tired. but content in the thought that one has accomplished some thing and advanced the whole work

towards its completion." Now, in the current hi-tech electronic era, working with language is just like playing game or planning a move on the chess board; where every individual tries to win it with application of maximum tricks and thoughts. The lexicographical work changes from being "tedious" to "enjoyable", when it runs on high performing computer system with the visual platform.

In fact, in today's fast developing world, the electronic application on language is a revolution. The natural language processing becomes enhanced in the application and machine implementation of syntactic, lexicographic, morphological analyses of language resulting in certain of NLP tools. The future of Computational Linguistics in India is hoped to be positive, with MT system-related work on lexicons, dictionaries, and thesauri already growing on in several centers of research. For instance, the two projects ANUSAARAKA at the Centre for Applied Linguistics and Translation Studies at Hyderabad in collaboration with the Indian Institute of Technology (IIT), Kanapur, and several other endeavours supported by the Department of Electronics (DoE), or the Central Government of India, are the best examples. The Computer Assisted Translation System (CATS), being developed by Centre for Developing Advanced Computing (C-DAC), Pune, is another instance of an NLP project contributing to lexicon build up. The lexicons, especially the semantic thesauri, developed in these centers, are sure to have a cascading effect on the other institutions or researchers producing similar products. The present work has been possible within the Applied Linguistics tradition at this point of time only because of the spread of the NLP research to the Indian languages in the last decade of this century. The programme developed here should not be taken as a final product but is rather a system which, through application and modification, will turn into a more viable commercially usable system.

REFERENCES

Andres, Susie. 1980. The computer, a tool for dictionary making in India. In: B. G. Misra. (ed.). **Lexicography in India.** Mysore: Central Institute of Indian Languages (CIIL). 145-156.

Bahn, Hardev. 1980. Hindi lexicography since independence and its feature. In: B. G. Misra. (ed.). **Lexicography in India.** Mysore: Central Institute of Indian Languages (CIIL).41-54.

Baker, Sheridan. 1972. The sociology of dictionaries and sociology of words. In: Howard D. Weinbrot (ed). **New Aspects of Lexicography : Literary Criticism, Intellectual History, and Social Change.** Carbondale and Edwardsville : Southern Illinois University Press. 138-151.

Batori, Istvan S., Winfried Lendres and Wolfgang Putschka (eds.). 1989. **Computational Linguistics : An International Handbook on Computer Oriented Language Research and Applications.** Berlin : Walter de Gruyter & Co.

Bolinger, Dwight. 1985. Defining the indefinable. In: C. J. Brumfit (Gen. ed.) **Dictionaries, Lexicography and Language Learning.** Oxford: Pergamon Press in association with The British Council. 69-73.

Boguraev, Bran and Ted Briscoe (eds.). 1989. Introduction. In: Bran Boguraev and Ted Briscoe (eds.). **Computational Lexicography for Natural Language Processing.** London and New York: Longman Group. 1-40.

Boguraev, B. (guest ed.). 1991. Building a lexicon. In: **International Journal of Lexicography.** UK:OUP. 4:3.

Gajendragadkar. S. N. 1973. Sanskrit historical dictionary and Vedanta. In : A.M. Ghatage *et al* (eds). **Studies in Historical Sanskrit Lexicography. Poona** : Deccan College. 13-24.

Garvin, Paul. L. 1966. Computer participation in linguistic research. In: **Language** 38. 385-389.

Ghatage, A. M. 1973. Traditional lexicons and Sanskrit historical dictionary. In : A.M. Ghatage *et al* **(eds). Studies in Historical Sanskrit Lexicography.** Poona: Deccan College. 25-39.

Grishman, Ralph. 1986. **Computational Linguistics: An Introduction.** Cambridge: Cambridge University Press.

Hartmann, R. R. K. 1983. On Theory-and Practice : Theory and practice in dictionary making. **In** : Hartmann R. R. K. (ed) **Lexicography: Principles and Practices.** 3-11.

Hayakawa, S. **I.** 1941. **Language in Thought and Action.** New York: Harcourt Brace and World **Inc.**

Hays, David G. 1967. **An Introduction to Computational Linguistics.** New York : American Elsevier Publishing Company, Inc.

Jackson, Howard. 1985. Grammar in the dictionary In: J. C. Brumfit (Gen ed) **Dictionaries, Lexicography and Language Learning.** Oxford: Pergamon Press in association with The British Council. 53-59.

Katre, Sumitra Mangesh. 1965. **Lexicography.** Annamalainagar: Annamalai University Publications in Linguistics No. 6.

Kelkar, Ashok R. 1973. The scope of historical dictionary. In : A.M. Ghatage *et al* (eds). **Studies in Historical Sanskrit Lexicography.** Poona : Deccan College. 57-69.

Knowles, F. 1983. Towards the machine dictionary : Mechanical dictionaries. In : Hartmann, R.R.K. **(ed.). Lexicography : Principles and Practice.** London : Academic Press, Inc. 181-193.

Kumar, Arvinda and Kusum Kumar. 1996. **sAmAntara kosha : Hindi thisArasa.** New Delhi: National Book Trust.

Laird, Charlton (Coml.). 1990. **Webster's New World Thesaurus.** (Rev. ed.). New York : Warner Books. Inc.

Levin, Beth. 1991. Building a lexicon : The contribution of linguistics. In: **International Journal of Lexicography.** UK : OUP. 4:3.205-226.

McGregor, Charles. 1985. From first idea to finished artefact : the general editor as chief engineer. In: C. J. Brumfit (Gen. ed). **Dictionaries, Lexicography and Language Learning.** Oxford: Pergamon Press in association with The British Council. 123-132.

McNaught, John. 1988. Computational lexicography and computational linguistics. In: **Lexicographica.** Tubingen: Max Niemeyer. 4. 19-33.

Mehendale. M. A. 1973. Analysis of meaning. In : A.M. Ghatage *et al* (eds). **Studies in Historical Sanskrit Lexicography.** Poona : Deccan College. 76-88.

Nagao, Makoto. 1989. **Machine Translation** : **How Far Can It** Go? Oxford : Oxford University Press.

Pandey, Mahendra Kumar C. 1993. Lexico-semantic space and transfiguration. A paper presented in the Second Bienniel Congress of Comparative Literature Association of India. Delhi : University of Delhi.

Pandey, Mahendra Kumar C. 1994. Discrepancy in semantic assignement in bilingual dictionary. In: **PILC Journal of Dravidian Studies.** Vol 4:2. July. Pondicherry: Pondicherry Institute of Linguistics and Culture. 161-172.

Pandey, Mahendra Kumar C. 1995. Semantic embodiment and Decipherment in "AmarkosrT. A **paper** presented in the **National Seminar on the Art of Dictionary Making in Ancient India.** CASS. University of Pune. 20-22 February, 1995.

Roget's Thesaurus of Synonyms & Antonyms 1988 (edn). USA: P.S.I. & Associates, Inc.

Sanjeevi, N. 1980. Some suggestions for the making of new dictionaries In: B. G. Misra. (ed.). **Lexicography in India**. Mysore: Central Institute of Indian Languages (CIIL). 126-130

Sarup, Laximan. 1967. **The nighaNTu and the niruktA: Text and Translation.** Delhi: Motilas Banarsidass.

Singh, Amar. 1995. **amarakoSaH: bhASA TIkA.** Varanasi: Chaukhamba Vidyabhavan.

Singh, Ram Adhar. 1982. **An Introduction to Lexicography**. Mysore: Central Institute of Indian Languages. CIIL Occasional Monograph Series No. 26.

Singh, Suraj Bhan. 1980. Concepts of semantic field and collocation in Hindi/Urdu lexicography. In: **Stvdia Et Acta Orientalia**. Bucarest.130-145

Singh, Udaya Narayana and Mahendra Kumar C. Pandey 1996. Semantic categories for electronic thesaurus for Indian languages. A paper presented in the **Indian Congress on Knowledge and Language.** CIIL, Mysore, 11-17 January, 1996.

Urdang, Lawrence (ed.).1991. **The Oxford Thesaurus : An A-Z Dictionary of Synoyms.** Oxford: Clarendon Press.

Webster's Collegiate Thesaurus. 1991. USA: Merriam Webster.

Zgusta, Ladislav. 1971. **Manual of Lexicography**. The Hague: Mouton.

APPENDIX 1

THES.MAK PROJECT : PACKAGE

AS

APPENDIX 1

THES.MAK PROJECT

IS

SUBMITTED

IN

1.44 MB FLOPPY- DISKETTE

PROCEDURES :

- Select your drive (e.g. C:\)
- Make Directory (e.g. C:\md Pandey)
- Copy **thes.exe** in the Directory (i.e. C:\ Pandey\...)
- Make Sub-Directory (e.g. C:\Pandey\md thesdata)
- Copy **lexis.mdb** & **lexis.ldb** in the Directory (e.g. C:\Pandey\thesmod\...)
- Run **thes.exe** from the ***C:\Pandey*** directory by typing **thes**

SYSTEM REQUIREMENT :

- Necessary DLLs or Visual Basic 5.0 and Access 2.0

APPENDIX 2

PROGRAMMING CODES FOR *THES.MAK* PROJECT

(Form-wise Code Representation)

FORM 1 : START

```
Private Sub Exit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub

Private Sub START_Click()
        'Display Search_form
        Form2.Visible = 1
        Form1.Visible = 1
End Sub

Private Sub StartInfo_Click()
        'Display InfoWindow
        Form1.Visible = 1
        Form4.Visible = 1
        Form2.Visible = 0
End Sub
```

FORM 2 : Search-Mode

```
Private Sub Exit_Click()
        Unload Form1
        Unload Form2
```

```
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub

Private Sub Info_Click()
        'Display the InfoWindow
        Form4.Visible = 1
        Form2.Visible = 1
End Sub

Private Sub lexical_Click()
        'Display Form5'
        Form6.Visible = 1
        Form2.Visible = 0
        Form1.Visible = 0
        Form6.HWEditbox.SetFocus
End Sub

Private Sub Semantic_Click()
        'Display Scr.Frm 3'
        Form3.Visible = 1
        Form2.Visible = 0
        Form1.Visible = 0
End Sub
```

FORM 3 : Semantic Domains

```
Private Sub Action_Click()
        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String
```

```
        WordListbox.Clear
        SCIListbox.Clear
        'Match ACTION
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "v" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SCIListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                                If Data1.Recordset(val3) = SCIListbox.List(j) Then
                                                        found = True
                                                        Exit For
                                                End If
                                        Next j
                                        If found = False Then
                                                SCIListbox.AddItem Data1.Recordset(val3)
                                                lIndex = lIndex + 1
                                        End If
                                End If
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        Action.Value = False
        Action.Enabled = True
End Sub

Private Sub Action_DblClick()
        'Clear ListBoxes
        SCIListbox.Clear
        SCIListbox.Clear

        'Match Action
```

```
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If  Data1.Recordset(val3) = "v" Then
                                found = True
                                WordListbox.AddItem Data1.Recordset("hw")
                        End If
                        If found = True Then
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub Affection_Click()
        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SClListbox.Clear

        'Match AFFECTION
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
```

```
                    val5 = CStr(val2)
                    val3 = val1 + val5
                    If Data1.Recordset(val3) = "a" Then
                        val3 = val4 + val5
                        If lIndex = 1 Then
                            found = False
                            SClListbox.AddItem Data1.Recordset(val3)
                            lIndex = lIndex + 1
                        Else
                            For j = lIndex - 1 To 0 Step -1
                                If Data1.Recordset(val3) = SClListbox.List(j) Then
                                    found = True
                                    Exit For
                                End If
                            Next j
                            If found = False Then
                                SClListbox.AddItem Data1.Recordset(val3)
                                lIndex = lIndex + 1
                            End If
                        End If
                    End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        Affection.Value = False
        Affection.Enabled = True
End Sub

Private Sub Affection_DblClick()

        'Clear ListBoxes
         WordListbox.Clear
         SClListbox.Clear

        'Match Affection
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
```

```
                    val3 = val1 + val4
                    If  Data1.Recordset(val3) = "a" Then
                            found = True
                            WordListbox.AddItem Data1.Recordset("hw")
                    End If
                    If  found = True Then
                            Exit For
                    End If
            Next i
            Data1.Recordset.MoveNext
    Loop
End Sub

Private Sub Back_Click()
    'Clear the boxes
    WordListbox.Clear
    SCIListbox.Clear

    'Display Scr.Frm2'
    Form3.Visible = 0
    Form1.Visible = 1
    Form2.Visible = 1
End Sub

Private Sub Exit_Click()
    Unload Form1
    Unload Form2
    Unload Form3
    Unload Form4
    Unload Form5
    Unload Form6
    Unload Form7
    Unload Form8
    Unload Form9
    Unload Form10
    Unload Form11
    Unload Form12
    Unload Form13
    Unload Form14
    Unload IAMP
    Unload SClass
    Unload SemClass
    Unload semclass1
    Unload SOPA
End Sub

Private Sub Info_Click()
    Form4.Visible = 1
    Form3.Visible = 1
```

```
End Sub

Private Sub Intellect_Click()
        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SCIListbox.Clear

        'Match INTELLECTION
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If  Data1.Recordset(val3) = "i" Then
                                val3 = val4 + val5
                                If  lIndex = 1 Then
                                        found = False
                                        SCIListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                           If  Data1.Recordset(val3) = SCIListbox.List(j) Then
                                                found = True
                                                Exit For
                                           End If
                                        Next j
                                        If found = False Then
                                                SCIListbox.AddItem Data1.Recordset(val3)
                                                lIndex = lIndex + 1
                                        End If
                                End If
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        Intellect.Value = False
```

```
        Intellect.Enabled = True
End Sub

Private Sub Intellect_DblClick()
        'Clear ListBoxes
        WordListbox.Clear
        SCIListbox.Clear

        'Match Intellection
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If Data1.Recordset(val3) = "i" Then
                                found = True
                                WordListbox.AddItem Data1.Recordset("hw")
                        End If
                        If found = True Then
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub lexscreen_Click()
        Form3.Visible = 0
        Form6.Visible = 1
End Sub

Private Sub ListHW_DblClick()
        'Clear ListBoxes
         WordListbox.Clear
        SCIListbox.Clear


        'set database
        Data1.DatabaseName = "thesdata\Lexis.mdb"
```

```
        Data1.RecordSource = "entry"
        Data1.Refresh
        Do While Data1.Recordset.EOF = False
                If Data1.Recordset.Fields("hw") <> "" Then
                        WordListbox.AddItem Data1.Recordset("hw")
                End If
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub Measure_Click()
        WordListbox.Clear
        SClListbox.Clear

        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String


        'Match MEASUREMENT
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "m" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SClListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                            If Data1.Recordset(val3) = SClListbox.List(j) Then
                                                        found = True
                                                        Exit For
                                            End If
                                        Next j
                                        If found = False Then
```

```
                                        SCIListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                End If
                        End If
                End If
        Next i
        Data1.Recordset.MoveNext
    Loop
    Measure.Value = False
    Measure.Enabled = True
End Sub

Private Sub Measure_DblClick()
    'Clear ListBoxes
    WordListbox.Clear
    SCIListbox.Clear

    'Match Measurement
    Data1.DatabaseName = "thesdata\Lexis.mdb"
    Data1.RecordSource = "entry"
    Data1.Refresh

    Do While Data1.Recordset.EOF = False
        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "sd"
        For i = 1 To 12 Step 1
            val2 = i
            val4 = CStr(val2)
            val3 = val1 + val4
            If Data1.Recordset(val3) = "m" Then
                found = True
                WordListbox.AddItem Data1.Recordset("hw")
            End If
            If found = True Then
                Exit For
            End If
        Next i
        Data1.Recordset.MoveNext
    Loop
End Sub


Private Sub other_Click()
    Dim val1 As String
    Dim val2 As Integer
```

```
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SClListbox.Clear

        'Match OTHER
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1

        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "o" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SClListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                            If Data1.Recordset(val3) = SClListbox.List(j) Then
                                                    found = True
                                                    Exit For
                                            End If
                                        Next j
                                        If found = False Then
                                            SClListbox.AddItem Data1.Recordset(val3)
                                            lIndex = lIndex + 1
                                        End If
                                End If
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        other.Value = False
        other.Enabled = True
End Sub

Private Sub other_DblClick()
```

```
        'Clear ListBoxes
        WordListbox.Clear
        SClListbox.Clear

        'Match Other
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If Data1.Recordset(val3) = "o" Then
                                found = True
                                WordListbox.AddItem Data1.Recordset("hw")
                        End If
                        If found = True Then
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub Percept_Click()
        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SClListbox.Clear

        'Match PERCEPTION
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
```

```
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "p" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SCIListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                            If Data1.Recordset(val3) = SCIListbox.List(j) Then
                                                    found = True
                                                    Exit For
                                            End If
                                        Next j
                                        If found = False Then
                                                SCIListbox.AddItem Data1.Recordset(val3)
                                                lIndex = lIndex + 1
                                        End If
                                End If
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        Percept.Value = False
        Percept.Enabled = True
End Sub

Private Sub Percept_DblClick()
        'Clear ListBoxes
        WordListbox.Clear
        SCIListbox.Clear

        'Match Perception
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
```

```
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If Data1.Recordset(val3) = "p" Then
                                found = True
                                WordListbox.AddItem Data1.Recordset("hw")
                        End If
                        If found = True Then
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub


Private Sub Progno_Click()

        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SClListbox.Clear

        'Match PROGNOSIS
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "pr" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SClListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
```

```
                        Else
                                For j = lIndex - 1 To 0 Step -1
                                   If Data1.Recordset(val3) = SClListbox.List(j) Then
                                                found = True
                                                Exit For
                                   End If
                                Next j
                                If found = False Then
                                        SClListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                End If
                        End If
                End If
        Next i
        Data1.Recordset.MoveNext
    Loop
    Progno.Value = False
    Progno.Enabled = True
End Sub

Private Sub Progno_DblClick()

    'Clear ListBoxes
    WordListbox.Clear
    SClListbox.Clear

    'Match Prognosis
    Data1.DatabaseName = "thesdata\Lexis.mdb"
    Data1.RecordSource = "entry"
    Data1.Refresh

    Do While Data1.Recordset.EOF = False
        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "sd"
        For i = 1 To 12 Step 1
            val2 = i
            val4 = CStr(val2)
            val3 = val1 + val4
            If Data1.Recordset(val3) = "pr" Then
                found = True
                WordListbox.AddItem Data1.Recordset("hw")
            End If
            If found = True Then
                Exit For
            End If
```

```
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub Reset_Click()
        WordListbox.Clear
        SClListbox.Clear
End Sub


Private Sub SClListbox_DblClick()
        WordListbox.Clear

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val As String
        val1 = "sc"
        Do While Data1.Recordset.EOF = False
                For i = 1 To 12 Step 1
                        val2 = i
                        val3 = CStr(val2)
                        val4 = val1 + val3
                        If SClListbox.List(SClListbox.ListIndex) = Data1.Recordset(val4) Then
                                WordListbox.AddItem Data1.Recordset("hw")
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub self_Click()
        Dim val1 As String
        Dim val2 As Integer
        Dim val5 As String
        Dim val3 As String
        Dim val4 As String

        WordListbox.Clear
        SClListbox.Clear

        'Match SELF
        Data1.DatabaseName = "thesdata\Lexis.mdb"
```

```
        Data1.RecordSource = "entry"
        Data1.Refresh
        lIndex = 1
        Do While Data1.Recordset.EOF = False
                found = False
                val1 = "sd"
                val4 = "sc"
                For i = 1 To 12 Step 1
                        found = False
                        val2 = i
                        val5 = CStr(val2)
                        val3 = val1 + val5
                        If Data1.Recordset(val3) = "s" Then
                                val3 = val4 + val5
                                If lIndex = 1 Then
                                        found = False
                                        SClListbox.AddItem Data1.Recordset(val3)
                                        lIndex = lIndex + 1
                                Else
                                        For j = lIndex - 1 To 0 Step -1
                                            If Data1.Recordset(val3) = SClListbox.List(j) Then
                                                    found = True
                                                    Exit For
                                            End If
                                        Next j
                                        If found = False Then
                                                SClListbox.AddItem Data1.Recordset(val3)
                                                lIndex = lIndex + 1
                                        End If
                                End If
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
        self.Value = False
        self.Enabled = True
End Sub

Private Sub Self_DblClick()
        'Clear ListBoxes
        WordListbox.Clear
        SClListbox.Clear

        'Match Self
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh

        Do While Data1.Recordset.EOF = False
```

```
                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sd"
                For i = 1 To 12 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If Data1.Recordset(val3) = "s" Then
                                found = True
                                WordListbox.AddItem Data1.Recordset("hw")
                        End If
                        If found = True Then
                                Exit For
                        End If
                Next i
                Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub WordListbox_DblClick()
        If WordListbox.List(WordListbox.ListIndex) <> "0" Then
                Form6.Visible = 1
                Form3.Visible = 0
                Form6.HWEditbox.Text = WordListbox.List(WordListbox.ListIndex)
        End If
End Sub
```

FORM 4 : INFO_WINDOW

```
Private Sub IWtxt_Click()
        'Display StartInfo_Form
        Form4.Visible = 1
        Form5.Visible = 1
End Sub

Private Sub Label1_Click()
        SClass.Visible = 1
        Form4.Visible = 0
End Sub


Private Sub LDtxt_Click()
        'Display Central_Form
        Form4.Visible = 1
        Form6.Visible = 1
```

```
End Sub


Private Sub LexDisInfotxt_Click()
        'Display Sem-Dom_form
        Form4.Visible = 1
        Form9.Visible = 1
End Sub

Private Sub SDtxt_Click()
        'Display Sem-Dom_form
        Form4.Visible = 1
        Form3.Visible = 1
End Sub

Private Sub SemDom_Info_Click()
        'Display SearchModeInfo_form
        Form8.Visible = 1
        Form4.Visible = 1
End Sub

Private Sub SMInfotxt_Click()
        'Display SearchModeInfo_form
        Form7.Visible = 1
        Form4.Visible = 1
End Sub

Private Sub SMtxt_Click()
        'Display Search_form
        Form2.Visible = 1
        Form4.Visible = 1
End Sub


Private Sub starttext_Click()
        'Display Start_form
        Form4.Visible = 1
        Form1.Visible = 1
End Sub
```

FORM 5 : Start_Info

```
Private Sub startinfoback_Click()
        Form4.Visible = 1
        Form5.Visible = 0
End Sub

Private Sub startinfoexit_Click()
```

```
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub
```

FORM 6 : LEXICAL DISPLAY

```
Private Sub Attributes_Click()
        Form11.Visible = 1
        Form10.Visible = 0
End Sub

Private Sub Attributes_DblClick()
        Form11.Visible = 1
        Form10.Visible = 0
End Sub


Private Sub ClearReset_Click()
        Titletext.Text = ""
        LSClistbox.Text = ""
        HWEditbox.Text = ""
        GCcombox.Clear
        SynListbox.Clear
End Sub

Private Sub Display_Click()
        Dim val As String
        val = HWEditbox.Text

        'Clear list box/comboboxh
        SynListbox.Clear
        GCcombox.Clear
```

```
        LSClistbox.Text = ""
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "gc"

        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                        Data1.Recordset("iw") Then
                        For i = 1 To 4 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If val <> "" Then
                                        If Data1.Recordset(val3) <> "" Then
                                                found = True
                                                GCcombox.AddItem Data1.Recordset(val3)
                                        End If
                                End If
                        Next i
                        Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If
        Loop
        Titletext.Text = "Synonym"
End Sub

Private Sub Form_Click()
        Form11.Visible = 0
End Sub

Private Sub GCcombox_Click()
        Dim val As String
        val = HWEditbox.Text

        'Clear Boxes
        SynListbox.Clear
        LSClistbox.Text = ""
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
```

```
'GCcombox noun display
Do While Data1.Recordset.EOF = False
        If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then

                found = False
                Dim val1 As String
                Dim val2 As Integer
                Dim val3 As String
                Dim val4 As String
                val1 = "sn"
                If GCcombox.Text = "saM" Then
                        For i = 1 To 12 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data1.Recordset(val3) <> "" Then
                                        found = True
                                        SynListbox.AddItem Data1.Recordset(val3)
                                End If
                        Next i
                End If
                'GCcombox verb display
                found = False
                Dim val5 As String
                Dim val6 As Integer
                Dim val7 As String
                Dim val8 As String
                val5 = "sn"
                If GCcombox.Text = "kri" Then
                        For i = 13 To 24 Step 1
                                val6 = i
                                val8 = CStr(val6)
                                val7 = val5 + val8
                                If Len(Data1.Recordset(val7)) = 0 Then
                                        found = True
                                        SynListbox.AddItem Data1.Recordset(val7)
                                End If
                        Next i
                End If
                'GCcombox adjective display
                found = False
                Dim val9 As String
                Dim val10 As Integer
                Dim val11 As String
                Dim val12 As String
                val9 = "sn"
                If GCcombox.Text = "vi" Then
                        For i = 25 To 36 Step 1
```

```
                        val10 = i
                        val12 = CStr(val10)
                        val11 = val9 + val12
                        If Data1.Recordset(val11) <> "" Then
                                found = True
                                SynListbox.AddItem Data1.Recordset(val11)
                        End If
                Next i
        End If
        'GCcombox adverb display
        found = False
        Dim val13 As String
        Dim val14 As Integer
        Dim val15 As String
        Dim val16 As String
        val13 = "sn"
        If GCcombox.Text = "krivi" Then
                For i = 37 To 48 Step 1
                        val14 = i
                        val16 = CStr(val14)
                        val15 = val13 + val16
                        If Data1.Recordset(val15) <> "" Then
                                found = True
                                SynListbox.AddItem Data1.Recordset(val15)
                        End If
                Next i
        End If
        Exit Do
    End If
    If val <> Data1.Recordset("hw") Then
        Data1.Recordset.MoveNext
    End If
  Loop
End Sub

Private Sub HWEditbox_Change()
  Dim val As String
  val = HWEditbox.Text
  Form10.Visible = 0
  Form11.Visible = 0

  SynListbox.Clear
  GCcombox.Clear
  LSClistbox.Text = ""
  Titletext.Text = ""
End Sub

Private Sub LBack_Click()
  HWEditbox.Text = ""
```

```
        LSClistbox.Text = ""
        Titletext.Text = ""
        SynListbox.Clear
        GCcombox.Clear
        Form6.Visible = 0
        Form1.Visible = 1
        Form2.Visible = 1
End Sub



Private Sub LExit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub

Private Sub lexscreen_Click()
        Form3.Visible = 1
        Form6.Visible = 0
        Form10.Visible = 0
        Titletext.Text = ""
        LSClistbox.Text = ""
        HWEditbox.Text = ""
        GCcombox.Clear
        SynListbox.Clear
End Sub

Private Sub LInfo_Click()
        'Display Info
        Form4.Visible = 1
        Form6.Visible = 1
```

```
        Form11.Visible = 0
End Sub

Private Sub SynListbox_Click()
        LSClistbox.Text = ""
        Noun Matching SD-SC
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "sn"
        If GCcombox.Text = "saM" Then
                For i = 1 To 4 Step 1
                        val2 = i
                        val4 = CStr(val2)
                        val3 = val1 + val4
                        If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val3) Then
                                LSClistbox.Text = Data1.Recordset("sc1")
                        End If
                 Next i
        End If
        Dim val5 As String
        Dim val6 As Integer
        Dim val7 As String
        Dim val8 As String
        val5 = "sn"
        If GCcombox.Text = "saM" Then
                For j = 5 To 8 Step 1
                        val6 = j
                        val7 = CStr(val6)
                        val8 = val5 + val7
                        If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val8) then
                                LSClistbox.Text = Data1.Recordset("sc2")
                        End If
                Next j
        End If
        Dim val9 As String
        Dim val10 As Integer
        Dim val11 As String
        Dim val12 As String
        val9 = "sn"
        If GCcombox.Text = "saM" Then
                For k = 9 To 12 Step 1
                    val10 = k
                    val11 = CStr(val10)
                    val12 = val9 + val11
                    If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val12) Then
                                LSClistbox.Text = Data1.Recordset("sc3")
                    End If
```

```
                Next k
End If
'Verb Matching SD-SC

Dim val_1 As String
Dim val_2 As Integer
Dim val_3 As String
Dim val_4 As String
val_1 = "sn"
If GCcombox.Text = "kri" Then
        For i = 13 To 16 Step 1
            val_2 = i
            val_4 = CStr(val_2)
            val_3 = val_1 + val_4
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val_3) Then
                    LSClistbox.Text = Data1.Recordset("sc4")
            End If
        Next i
End If
Dim val_5 As String
Dim val_6 As Integer
Dim val_7 As String
Dim val_8 As String
val_5 = "sn"
If GCcombox.Text = "kri" Then
        For j = 17 To 20 Step 1
            val_6 = j
            val_7 = CStr(val_6)
            val_8 = val_5 + val_7
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val_8) Then
                    LSClistbox.Text = Data1.Recordset("sc5")
            End If
        Next j
End If
Dim val_9 As String
Dim val_10 As Integer
Dim val_11 As String
Dim val_12 As String
val_9 = "sn"
If GCcombox.Text = "kri" Then
        For k = 21 To 24 Step 1
            val_10 = k
            val_11 = CStr(val_10)
            val_12 = val_9 + val_11
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val_12) Then
                    LSClistbox.Text = Data1.Recordset("sc6")
            End If
        Next k
End If
```

```
'Adjective Matching SD-SC

Dim val1_1 As String
Dim val1_2 As Integer
Dim val1_3 As String
Dim val1_4 As String
val1_1 = "sn"
If GCcombox.Text = "vi" Then
        For i = 25 To 28 Step 1
            val1_2 = i
            val1_4 = CStr(val1_2)
            val1_3 = val1_1 + val1_4
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val1_3) Then
                    LSClistbox.Text = Data1.Recordset("sc7")
            End If
        Next i
End If

Dim val1_5 As String
Dim val1_6 As Integer
Dim val1_7 As String
Dim val1_8 As String
val1_5 = "sn"
If GCcombox.Text = "vi" Then
        For j = 29 To 32 Step 1
            val1_6 = j
            val1_7 = CStr(val1_6)
            val1_8 = val1_5 + val1_7
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val1_8) Then
                LSClistbox.Text = Data1.Recordset("sc8")
            End If
        Next j
End If
Dim val1_9 As String
Dim val1_10 As Integer
Dim val1_11 As String
Dim val1_12 As String
val1_9 = "sn"
If GCcombox.Text = "vi" Then
        For k = 33 To 36 Step 1
            val1_10 = k
            val1_11 = CStr(val1_10)
            val1_12 = val1_9 + val1_11
            If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val1_12) Then
                LSClistbox.Text = Data1.Recordset("sc9")
            End If
        Next k
End If
'Adverb Matching SD-SC
```

```
        Dim val2_1 As String
        Dim val2_2 As Integer
        Dim val2_3 As String
        Dim d As String
        val2_1 = "sn"
        If GCcombox.Text = "krivi" Then
                For i = 37 To 40 Step 1
                    val2_2 = i
                    val2_4 = CStr(val2_2)
                    val2_3 = val2_1 + val2_4
                    If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val2_3) Then
                        LSClistbox.Text = Data1.Recordset("sc10")
                    End If
                Next i
        End If
        Dim val2_5 As String
        Dim val2_6 As Integer
        Dim val2_7 As String
        Dim val2_8 As String
        val2_5 = "sn"
        If GCcombox.Text = "krivi" Then
                For j = 41 To 44 Step 1
                    val2_6 = j
                    val2_7 = CStr(val2_6)
                    val2_8 = val2_5 + val2_7
                    If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val2_8) Then
                            LSClistbox.Text = Data1.Recordset("sc11")
                    End If
                Next j
        End If
        Dim val2_9 As String
        Dim val2_10 As Integer
        Dim val2_11 As String
        Dim val2_12 As String
        val2_9 = "sn"
        If GCcombox.Text = "krivi" Then
                For l = 45 To 48 Step 1
                    val2_10 = l
                    val2_11 = CStr(val2_10)
                    val2_12 = val2_9 + val2_11
                    If SynListbox.List(SynListbox.ListIndex) = Data1.Recordset(val2_12) Then
                                LSClistbox.Text = Data1.Recordset("sc12")
                    End If
                Next l
        End If
End Sub

Private Sub SynListbox_DblClick()
        LSClistbox.Text = ""
```

```
        HWEditbox.Text = SynListbox.List(SynListbox.ListIndex)
End Sub
```

FORM 7 : SearchMode_Info

```
Private Sub searchinfoback_Click()
        Form4.Visible = 1
        Form7.Visible = 0
End Sub


Private Sub searchinfoexit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub
```

FORM 8 : SemDom_Info

```
Private Sub sdinfoback_Click()
        Form8.Visible = 0
        Form4.Visible = 1
End Sub


Private Sub sdinfoexit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
```

```
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub
```

FORM 9 : LEXDISPLAY INFO

```
Private Sub lexdisinfiexit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub

Private Sub lexdisinfoback_Click()
        Form9.Visible = 0
        Form4.Visible = 1
End Sub
```

FORM 10 : DIE

```
Private Sub defination_Click()
```

```
        Dim val As String
        Dim found As Boolean
        Dim blank As Boolean

        val = Form6.HWEditbox.Text

        DIEListbox.Clear

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "ide"
        Data1.Refresh

        found = False
        Data1.Recordset.MoveFirst
        Do While Data1.Recordset.EOF = False
            If Trim(val) = Data1.Recordset("hw") Or Trim(val) =
            Form6.Data1.Recordset("sv1") Or Trim(val) = Form6.Data1.Recordset("iw") Then
                    found = True
                    If Data1.Recordset.Fields("defin") <> "" Then
                            blank = False
                    Else
                            blank = True
                    End If
                    If found Then
                            If blank Then
                                    MsgBox "Definition is not available"
                                    Exit Do
                            Else
                                    DIEListbox.AddItem Data1.Recordset.Fields("defin")
                                    End If
                            End If
                End If
                Data1.Recordset.MoveNext
        Loop
        DIETitlebox.Text = "Definition"
End Sub

Private Sub DIEExit_Click()
        DIEListbox.Clear
        Form10.Visible = 0
End Sub

Private Sub Example_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        DIEListbox.Clear
        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "ide"
```

```
        Data1.Refresh

        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "examp"

        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Form6.Data1.Recordset("sv1") Or val =
                Form6.Data1.Recordset("iw") Then
                        For i = 1 To 12 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data1.Recordset(val3) <> "" Then
                                        If val = Data1.Recordset("hw") Then
                                                found = True
                                                DIEListbox.AddItem Data1.Recordset(val3)
                                        End If
                                End If
                        Next i
                        If Not found Then
                                MsgBox "Example is not available", vbOKOnly
                        End If
                        If val <> Data1.Recordset("hw") Then
                                SynListbox.AddItem Data1.Recordset("hw")
                        End If
                        Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If
        Loop
        DIETitlebox.Text = "Example"
End Sub

Private Sub illustration_Click()
        Dim val As String
        val = Form6.HWEditbox.Text
        DIEListbox.Clear

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "ide"
        Data1.Refresh

        found = False
        Dim val1 As String
```

```
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "illus"

        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Form6.Data1.Recordset("sv1") Or val =
                Form6.Data1.Recordset("iw") Then
                        For i = 1 To 12 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data1.Recordset(val3) <> "" Then
                                        If val = Data1.Recordset("hw") Then
                                                found = True
                                                DIEListbox.AddItem Data1.Recordset(val3)
                                        End If
                                End If
                        Next i
                        If Not found Then
                                MsgBox "Illustration is not available", vbOKOnly
                        End If
                        If val <> Data1.Recordset("hw") Then
                                SynListbox.AddItem Data1.Recordset("hw")
                        End If
                        Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If
        Loop
        DIETitlebox.Text = "Illustration"
End Sub
```

FORM 11 : Attributes

```
Private Sub anto_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        Rem Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Dim val1 As String
```

```
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "ant"
        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
                If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        For i = 1 To 12 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data2.Recordset(val3) <> "" Then
                                        found = True
                                        Form6.SynListbox.AddItem Data2.Recordset(val3)
                                End If
                        Next i
                        If Not found Then
                                MsgBox "Antonym is not available", vbOKOnly
                        End If
                        Exit Do
                End If
                If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <>
                Data1.Recordset("iw") Then
                        Data1.Recordset.MoveNext
                        Data2.Recordset.MoveNext
                End If
        Loop
        Form6.Titletext.Text = "Antonym"
End Sub

Private Sub collocation_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        Rem Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "coll"
        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
```

```
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
                If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        For i = 1 To 4 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data2.Recordset(val3) <> "" Then
                                        found = True
                                        Form6.SynListbox.AddItem Data2.Recordset(val3)
                                End If
                        Next i
                        If Not found Then
                                MsgBox "Collocation is not available", vbOKOnly
                        End If
                        Exit Do
                End If
                If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <
                Data1.Recordset("iw") Then
                        Data1.Recordset.MoveNext
                        Data2.Recordset.MoveNext
                End If
        Loop
        Form6.Titletext.Text = "Collocation"
End Sub


Private Sub Compound_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        Rem Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "com"
        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
                If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        For i = 1 To 5 Step 1
                                val2 = i
```

```
                    val4 = CStr(val2)
                    val3 = val1 + val4
                    If Data2.Recordset(val3) <> "" Then
                        found = True
                        Form6.SynListbox.AddItem Data2.Recordset(val3)
                    End If
                Next i
                If Not found Then
                    MsgBox "Compund word is not available", vbOKOnly
                End If
                Exit Do
            End If
            If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <>
                Data1.Recordset("iw") Then
                Data1.Recordset.MoveNext
                Data2.Recordset.MoveNext
            End If
        Loop
        Form6.Titletext.Text = "Compound"
End Sub

Private Sub dervat_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        Rem Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "drv"
        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
            If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
            Data1.Recordset("iw") Then
                For i = 1 To 5 Step 1
                    val2 = i
                    val4 = CStr(val2)
                    val3 = val1 + val4
                    If Data2.Recordset(val3) <> "" Then
                        found = True
                        Form6.SynListbox.AddItem Data2.Recordset(val3)
                    End If
```

```
                    Next i
                    If Not found Then
                        MsgBox "Derivational  word is not available", vbOKOnly
                    End If
                    Exit Do
                End If
                If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <>
                Data1.Recordset("iw") Then
                    Data1.Recordset.MoveNext
                    Data2.Recordset.MoveNext
                End If
        Loop
        Form6.Titletext.Text = "Derivative"
End Sub

Private Sub DIE_Click()

        Form10.DIEListbox.Clear

        Dim val As String
        val = Form6.HWEditbox.Text

        Data1.DatabaseName = "thesdata\Lexis.mdb"
        Data1.RecordSource = "entry"
        Data1.Refresh
        Data1.Recordset.MoveFirst
        Do While Data1.Recordset.EOF = False
            If Form6.HWEditbox.Text = Data1.Recordset("hw") Then
                Form10.Visible = 1
                Exit Do
            Else
                Form10.Visible = 0
            End If
            Data1.Recordset.MoveNext
        Loop
End Sub

Private Sub IdentWord_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        'Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Do While Data1.Recordset.EOF = False
```

```
                If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                    If Data1.Recordset("iw") <> "" Then
                        If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Then
                            found = True
                            Form6.SynListbox.AddItem Data1.Recordset("iw")
                        End If
                    End If
                    If val <> Data1.Recordset("hw") Then
                        Form6.SynListbox.AddItem Data1.Recordset("hw")
                    End If
                    Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                    Data1.Recordset.MoveNext
                End If
        Loop
        If Not found Then
                MsgBox "Identical Word is not available", vbOKOnly
        End If
        Form6.Titletext.Text = "Identical Word"
End Sub

Private Sub RelatWord_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        Rem Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        found = False
        Dim val1 As String
        Dim val2 As Integer
        Dim val3 As String
        Dim val4 As String
        val1 = "rw"
        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
                If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        For i = 1 To 12 Step 1
                                val2 = i
                                val4 = CStr(val2)
                                val3 = val1 + val4
                                If Data2.Recordset(val3) <> "" Then
                                        found = True
```

```
                                Form6.SynListbox.AddItem Data2.Recordset(val3)
                            End If
                        Next i
                        Exit Do
                End If
                If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <>
                Data1.Recordset("iw") Then
                        Data1.Recordset.MoveNext
                        Data2.Recordset.MoveNext
                End If
                Loop
                Form6.Titletext.Text = "Related Word"
End Sub


Private Sub Root_Click()
        Dim val As String
        val = Form6.HWEditbox.Text

        'Clear list box
        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""

        Data1.Recordset.MoveFirst
        Data2.Recordset.MoveFirst
        Do While Data2.Recordset.EOF = False Or Data1.Recordset.EOF = False
                If val = Data2.Recordset("hw") Or val = Data1.Recordset("sv1") Or val =
                Data1.Recordset("iw") Then
                        Form6.SynListbox.AddItem Data2.Recordset("root")
                        Exit Do
                End If
                If val <> Data2.Recordset("hw") Or val <> Data1.Recordset("sv1") Or val <>
                Data1.Recordset("iw") Then
                        Data1.Recordset.MoveNext
                        Data2.Recordset.MoveNext
                End If
        Loop
        Form6.Titletext.Text = "Root"
End Sub

Private Sub spllvar_Click()
        Dim val As String
        val = Form6.HWEditbox.Text
        'Clear list box

        Form6.SynListbox.Clear
        Form6.GCcombox.Clear
        Form6.LSClistbox.Text = ""
```

```
        found = False
        Data1.Recordset.MoveFirst
        Do While Data1.Recordset.EOF = False
                If val = Data1.Recordset("hw") Or val = Data1.Recordset("sv1") Then
                    If Data1.Recordset("sv1") <> "" And val = Data1.Recordset("hw") Then
                        found = True
                        Form6.SynListbox.AddItem Data1.Recordset("sv1")
                    End If
                    If Data1.Recordset("sv1") <> "" And val <> Data1.Recordset("hw")  then
                        found = True
                        Form6.SynListbox.AddItem Data1.Recordset("hw")
                    End If
                    Exit Do
                End If
                If val <> Data1.Recordset("hw") Then
                        Data1.Recordset.MoveNext
                End If
        Loop
        If Not found Then
                MsgBox "Spelling Variation is not available", vbOKOnly
        End If
        Form6.Titletext.Text = "Spell Variation"
End Sub
```

FORM 12: Info_Commands

```
Private Sub infocomattri_Click()
        Form13.Visible = 1
End Sub

Private Sub infocomdie_Click()
        Form14.Visible = 1
End Sub

Private Sub infocomexit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
```

```
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub

Private Sub infocomsearch_Click()
        Form7.Visible = 1
End Sub

Private Sub infocomsemdom_Click()
        Form8.Visible = 1
End Sub

Private Sub infocomstart_Click()
        Form5.Visible = 1
End Sub

Private Sub infoconlexdis_Click()
        Form9.Visible = 1
End Sub
```

FORM 13 : Attribute_Info

```
Private Sub attriinfoback_Click()
        Form4.Visible = 1
        Form13.Visible = 0
End Sub

Private Sub attriinfoexit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
```

```
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub
```

FORM 14 : DIE_Info

```
Private Sub dieinfoback_Click()
        Form4.Visible = 1
        Form14.Visible = 0
End Sub

Private Sub dieinfoexit_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub
```

FORM 15 (Sclass) : Semantic Scheme

```
Private Sub Label1_Click()
        SOPA.Visible = 1
        IAMP.Visible = 0
End Sub

Private Sub Label1_DblClick()
        SemClass.Visible = 1
        IAMP.Visible = 0
        SOPA.Visible = 0
        semclass1.Visible = 0
End Sub
```

```
Private Sub Label2_Click()
        IAMP.Visible = 1
        SOPA.Visible = 0
End Sub

Private Sub Label2_DblClick()
        semclass1.Visible = 1
        IAMP.Visible = 0
        SOPA.Visible = 0
        SemClass.Visible = 0
End Sub

Private Sub Label3_Click()
        Unload Form1
        Unload Form2
        Unload Form3
        Unload Form4
        Unload Form5
        Unload Form6
        Unload Form7
        Unload Form8
        Unload Form9
        Unload Form10
        Unload Form11
        Unload Form12
        Unload Form13
        Unload Form14
        Unload IAMP
        Unload SClass
        Unload SemClass
        Unload semclass1
        Unload SOPA
End Sub
```

FORM 16 (SemClass) : Semantic Domain 1-4 : Classes

```
Private Sub Label1_Click()
        SClass.Visible = 1
        SemClass.Visible = 0
End Sub
```

FORM 17 (semclass1) : Semantic Domain 5-8 : Classes

```
Private Sub Label1_Click()
        SClass.Visible = 1
        semclass1.Visible = 0
End Sub
```

APPENDIX - 3

A List of Headwords

(The following list of headwords is automatically sorted out as appeared in word list-box of the Semantic Domains screen.)

1. A~kaRA
2. A~kha jhapakate
3. A~khO~ dekhete
4. AbAda
5. abahulatA
6. AbharaNa
7. abhAva
8. abhAvagrastatA
9. abhilAva
10. abhimaMcana
11. abhinayana
12. abhishankA
13. abhishApa
14. abhishapta
15. AbhUSaNa
16. acaitanya
17. acambhA
18. AcAra
19. AcAra bhraSTa
20. AcAra shunya
21. AcAra vyavahAra
22. AcArabhraSTatA
23. AcArahIna
24. acaraja
25. acarajabharA
26. AcaraNa
27. AcaraNa shailI
28. AcaraNahIna
29. AcaraNahInatA
30. AcaraNashIla
31. AcAravAna
32. acetanA
33. acetanatA
34. adAvata
35. adbhuta
36. adhama
37. adharma
38. adharma karma
39. adharmI
40. adhiloka
41. adhivAsa
42. adhivAsI
43. adhivAsitA
44. adhogati
45. adhomArga
46. adhUrApana
47. agala bagala
48. agha
49. aghI
50. agraja
51. agrASana
52. ahasAsa
53. Ahuti
54. AjnAkAritA
55. ajnAnatA
56. ajnAnatA
57. ajnAnI
58. ajnAtA
59. ajnAna
60. akAlpanikatA
61. AkAsha
62. akhanDatA
63. akhila vishva
64. akhilatA
65. akla
66. aklamaMda
67. aklamaMdI
68. akshatatA
69. alaMkAra
70. alaukika ghaTanA
71. alaukika krita
72. Ama rAstA
73. amIra
74. amIra umarA
75. amIrI
76. amithyA
77. aMshahInatA
78. aMshatA
79. AMshika
80. AMshikatA
81. aMshIya
82. aMshIyatA
83. aMta
84. aMtariksha
85. AnA-jAnA
86. anabana
87. anAcAra
88. anAcArI
89. anAcAriNI
90. anAcAritA
91. anAja
92. anaMshatA
93. anaMta vishvIya
94. Anana phAnana
95. anAnshikatA
96. ananubhUti
97. andAja
98. anka
99. anna
100. annotpAdana
101. anubhUti
102. anubhUtihInatA
103. anugra
104. anugratA
105. anuja
106. anumAna
107. anupAlaka
108. anupAlana
109. anurAga
110. anushAsI
111. anushAsita
112. anuvaMsha shriMkhalA
113. apacArI
114. apadhvansha
115. apakArI
116. apakartA
117. apamArgI
118. apApa
119. aparAdha

120.apracuratA
121.apraja
122.apUrNa
123.apUrNAMsha
124.apUrNAMshatA
125.apUrNAMshatA
126.apUrNAMshika
127.apUrNatA
128.apUta
129.apUtA
130.ArAdhanA
131.araNya
132.AraNya
133.AraNyaka
134.araNyIya
135.arcanA
136.ari
137.aRosa paRosa
138.arthavattA
139.Asa paRosa
140.Asa pAsa
141.asadAcAra
142.asadAcArI
143.asadAcAriNI
144.asAdhu
145.asAdhutA
146.asajjanatA
147.asala me~
148.asaliyata
149.asamagra
150.asamagratA
151.AsamAna
152.asamasta
153.asamastatA
154.asampannatA
155.asampUrNa
156.asampUrNatA
157.asamriddhatA
158.asamriddhi
159.asaMtAna
160.asarvAMgatA
161.asarVaMsha
162.as̃akala
163.as̃akalatA
164.ashaktatA
165.ashakti
166.AshaMkA
167.Ashcarya
168.Ashcaryajanaka
169.AshcaryapUrNa
170.AshcaryapUrNa
171.asheshatA
172.AshIrvacana
173.AshIrvAda
174.AshISa
175.AsIsa
176.asIsa
177.asta
178.AsthA
179.aTakala
180.ateja
181.atishIghra
182.Atmaja
183.AtmajA
184.Atmanigraha
185.AtmanigrahI
186.AtmaniyantraNa
187.Atmaniyantrita
188.AtmAnushasana
189.AtmAnushAsI
190.AtmAnushAsita
191.AtmasaiMyama
192.aulAda
193.AvAgamana
194.AvAjAhI
195.avaMsa
196.AvArA
197.AvAsa
198.avashapta
199.AvAsI
200.AvAsitA
201.avataMsa
202.avibhAjitatA
203.avikalatA
204.avikhaMDitatA
205.avilamba
206.avishvAsI
207.azIza
208.azIzI
209.bacapana
210.bacce
211.bada
212.badacalana
213.badacalanI
214.badakAra
215.badakArI
216.badamAsha
217.badamAshI
218.badana
219.badanIyatI
220.baira
221.bairI
222.bAla gopAla
223.bAlabacce
224.balahInatA
225.bAlaka
226.bAlyAvasthA
227.baMdha
228.baMdhana
229.baMdhu
230.baMtAdhAra
231.banailA
232.banakhaMDI
233.banavAsI
234.barabADI
235.barasAta
236.baratana
237.baratAva
238.baRe loga
239.barRe bhAI
240.basara
241.bAshindA
242.bATa
243.beakklI
244.beaulAda
245.becirAga
246.behoshI
247.beRA
248.besudhI
249.beTA
250.beTA beTI
251.beTavA
252.beTI
253.bevakUpha
254.bevakUphI
255.bhadra
256.bhAgika
257.bhAI
258.bhAIbaMdhu
259.bhaiyA
260.bhAjI
261.bhakta
262.bhakti bhAva
263.bhalA
264.bhalamanasAhata
265.bhalamanasI
266.bhalApana

267.bharaNa poSaNa
268.bhartA
269.bhASA
270.bhASA-bhASI
271.bhASI
272.bhASitA
273.bhava
274.bhAvanA
275.bhaviSya draSTA
276.bhaviSya kathana
277.bhaviSya vaktA
278.bhaviSya vANI
279.bhodUpanA
280.bhoga
281.bhraSTa
282.bhraSTA
283.bhraSTAcAra
284.bhraSTAcArI
285.bhraSTatA
286.bhrAtA
287.bhU
288.bhUmi
289.bhUmigata mArga
290.bhUSaNa
291.bhuvana
292.bhuvanIya
293.bigaRA
294.bipannatA
295.biTiyA
296.biyAbAna
297.bodha
298.bodhahInatA
299.bolI
300.bolI-bhASA
301.brahamAnDa
302.brahamAnDIya
303.brAhma
304.buddhi
305.buddhihInatA
306.buddhimAna
307.buddhimAnI
308.buddhimattA
309.buddhUpanA
310.burA
311.burAI
312.buRhAI
313.buRhApA
314.bURhApana
315.buRhautI
316.cakarAnA
317.cAla bAjI
318.cala basanA
319.cAla calana
320.cAlAkI
321.cAma
322.camaRA
323.camaRI
324.camatkAra
325.camatkAraka
326.camatkArapUrNa
327.camatkArI
328.cAmatkArika
329.camatkriti
330.caRhata
331.caRhAvA
332.caritra
333.caritrahIna
334.caritrahInA
335.caritrahInatA
336.caritravAna
337.caritravatI
338.caritrI
339.carma
340.cAruSIla
341.cAruSIlA
342.caTAka paTAka
343.caTApaTa
344.caTapaTa
345.caturthAnsha
346.cautha
347.cauthA bhAga
348.cauthA hissA
349.cauthApana
350.cetanA
351.cetanahInatA
352.chAla
353.chalapUrNatA
354.chIchAledara
355.choTe
356.choTe bhAI
357.ciRiyA
358.cokhA
359.cukanA
360.cuTakI bajAte
361.DagamagAnA
362.Dagara
363.dana se
364.danAdana
365.Dangara
366.dara-asala
367.darshana icchuka
368.darshanAbhilASI
369.darshanArthI
370.daulata
371.daulatamaMda
372.daulatamaMdI
373.davArpaNa
374.deha
375.dehAvaraNa
376.dekhabhAla
377.dekhabhAla karanA
378.dekhate dekhate
379.deva darshaka
380.devAhuti
381.dhairya
382.dhana daulata
383.dhanadhAnya
384.dhanadhAnyahInatA
385.dhanadhAnyapUrNatA
386.dhanADhya
387.dhanADhya jana
388.dhanADhyatA
389.dhanapAla
390.dhanavAna
391.dhandhA
392.dhanI
393.dhanI jana
394.dhanika
395.dhanikatA
396.dhannA
397.dhAnya
398.dhaRAka se
399.dharatI
400.dharma goSThI
401.dhIra
402.dhIraja
403.Dhora
404.dhUrtatA
405.dhvaMsha
406.dishmanAI
407.Do~gA
408.DolanA
409.dolita honA
410.dosta
411.dostAnA
412.dostI
413.dostI yArI

414.droha
415.drohI
416.duA
417.duniyA
418.durAcAra
419.durAcArI
420.durAcAriNI
421.durAcaritA
422.duravasthA
423.durdashA
424.durgata
425.durgati
426.durita
427.durjanatA
428.dushcaritra
429.dushcaritrA
430.dushcaritratA
431.dushkarmA
432.dushkarma
433.dushmana
434.dushmanI
435.dushshIla
436.duSTA
437.duSTa
438.duSTa AcaraNa
439.duSTAcArI
440.duSTAcAriNI
441.duSTatA
442.duSTAtmA
443.dveSI
444.gA~va
445.ga~vaI
446.ga~vAra
447.gA~vavAlA
448.gahanA
449.galata rAstA
450.gaII
451.gaII kUcA
452.gallA
453.gAma nivAsI
454.gamana Agamana
455.gamARiyA
456.gambhIra
457.gambhIratA
458.gaMdA
459.gata
460.ghara
461.ghara-bAra
462.gharAnA
463.gharelU
464.ghUmanA
465.ghUmanA-phiranA
466.girA
467.giri
468.giricara
469.girijana
470.gota
471.gotra
472.grAma
473.grAmavAsI
474.grAmiNa
475.grAmya
476.griha
477.grihadIpti
478.grihashobhA
479.grihasthi
480.guMDA
481.guMDA liccA
482.guMDAgarda
483.guMDaI
484.guMDApana
485.gunAha
486.guNavAna
487.gunDAgardI
488.gupta mArga
489.guzArA
490.guzara basara
491.hairAnI
492.hairata
493.hakIkata
494.hakIkata me~
495.harAmagirI
496.harAmakArI
497.herApherI
498.hilanA
499.hilanA-DulanA
500.hunara
501.ImAna
502.irdagirda
503.ishka
504.iti
505.jAduI
506.jagatIya
507.jahA~
508.jala
509.jala kUpa
510.jalAshaya
511.jalda
512.jaldI
513.jaldI se
514.jaMgaII
515.jamI~dAra
516.jamI~dArI
517.jamIna
518.janamArga
519.jananI
520.janapatha
521.jAnavara
522.jangala
523.janmadAtrI
524.janmadAyinI
525.jarA
526.jarA sA
527.jaRatA
528.jaraThatA
529.jAtA
530.jAtaka
531.jAti
532.javA~
533.javAna
534.javAnI
535.jAyadAda
536.jhannATe se
537.jhapATe se
538.jhaTa
539.jhaTaka paTAka
540.jhaTAka se
541.jhaTapaTa
542.jIrNatA
543.jIvana nirvAha
544.jIvana yApana
545.jnAna
546.jnAnavAna
547.jogI
548.jota
549.jyotiSI
550.kabIlA
551.kadAcAra
552.kadAcArI
553.kadAcAriNI
554.kadAcAritA
555.kahate kahate
556.kaIA
557.kaluSA
558.kaluSita
559.kama mAtrA me~
560.kamAla

561.kamatAI
562.kamazorI
563.kamI
564.kamInagI
565.kamInApana
566.kAnana
567.kanyA
568.kanyA ratna
569.kapaRA
570.kapaTa
571.kapaTatA
572.kAphI
573.kapota
574.kArabAra
575.kaRakI
576.karAmatI
577.karataba
578.karishmA
579.karma kalApa
580.kArobAra
581.kAshtakAra
582.kAshtakArI
583.kashtI
584.kaTAI
585.kaTarA
586.kAyA
587.keveTAI
588.khAdyAnna
589.khaga
590.khAla
591.khalAcAra
592.khalatA
593.khAnadAna
594.khanDahInatA
595.khanDatA
596.khanDIya
597.khanDIyatA
598.khapanA
599.kharAba
600.khaTa se
601.khaTAka se
602.khAtmA
603.khazAnA
604.khetI
605.khetI bARI
606.khetIhara
607.khevAI
608.khevana
609.khilAnA pilAnA
610.khisakanA
611.khivAI
612.khoTAI
613.khoTApana
614.khulA mArga
615.khUTanA
616.killata
617.kisAna
618.kisAnI
619.kishtI
620.koliyA
621.koliyA kaTarA
622.kotAhI
623.kriSaka
624.kriSi
625.kshaNa
626.kshaya
627.kshINatA
628.kU~A
629.ku~ara
630.ku~vara
631.kUcA
632.kUcA kaTarA
633.kukarmI
634.kukarmiNI
635.kula
636.kuladhara
637.kulandhara
638.kuliyA
639.kumArga
640.kumArgI
641.kumArgiNI
642.kUpa
643.kupanthI
644.kupatha
645.kUTatA
646.kuTilatA
647.kutsita
648.kutsitatA
649.kutUhala
650.kutUhalapUrNa
651.kuvritti
652.lakshmIvAna
653.lAlana pAlana
654.lalita kalA
655.lARa-pyAra
656.laRakA laRakI
657.laRakapana
658.laRakI
659.launI
660.lAva
661.lavanI
662.laya
663.livAsa
664.lopa
665.luccAI
666.lunAI
667.mA~
668.mA~jhIgIrI
669.maga
670.mahAvishvavIya
671.mAhIgIrI
672.maitrI
673.maitrIpUrNa
674.makAna
675.mAla
676.mAla matA
677.mAla TAla
678.mAladAra
679.mAlAmAla
680.maMca prastutIkaraNa
681.maMcana
682.maMcIkaraNa
683.manamIta
684.manonubhUti
685.mApa
686.maraNa
687.mArga
688.maryAdAshIlatA
689.mAtA
690.mauta
691.maveshI
692.mela
693.milakIyata
694.mIta
695.mitAI
696.mitavA
697.mitra
698.mitra bhAva
699.mitra sambaMdha
700.mitratA
701.mitratApUrNa
702.mitravara
703.mitrocita
704.mleccha
705.mrityu
706.muddaI
707.muhabbata

708.mukaranA
709.muni
710.muRanA
711.mURhatA
712.mUrkha
713.mUrkhatA
714.mUrkhatA
715.nAca
716.nAca gAnA
717.nAdAnI
718.naivedya
719.naiyA
720.najadIkI
721.nambara
722.nApa
723.nApanA
724.narAdhama
725.narakIya
726.nAsamajhI
727.nAsha
728.nATaka maMcana
729.nATya prastuti
730.naujavAna
731.naukA
732.naukAcAlana
733.naukarma
734.naukAyana
735.naukhila
736.nAva
737.nAvika karma
738.neka
739.neka calana
740.nekanIyatI
741.nekI
742.nestanAbUdI
743.nIca
744.nIcatA
745.nigama
746.nigraha
747.nigrahI
748.nigrahita
749.nikAI
750.nikaTatA
751.nikhila
752.nikhilatA
753.nikhuTanA
754.nimeSa
755.nipUtA
756.nipUta
757.nIra
758.nirabaMshiyA
759.nirvAhita
760.nirvaMsha
761.nirvaMshatA
762.niSpApatA
763.nissantAna
764.nissantAnatA
765.niSThA
766.nivAsa
767.nivAsa-sthAna
768.nivAsI
769.nivAsitA
770.niyamI
771.niyatAtmA
772.nritya
773.pAbaMdI
774.pAgalapana
775.pahanAva
776.pahanAvA
777.pahARa
778.paharAvA
779.pahARI
780.pahARiyA
781.paisA vAlA
782.pAjIpanA
783.pAka
784.pakherU
785.pakSI
786.palA
787.pala
788.pala me~
789.pAla posa
790.pAlA posA
791.palabhara me~
792.pAlaka
793.palaka jhapakate
794.pAlaka poSaka
795.pAlana
796.pAlanA
797.pAlana karanA
798.pAlana kartA
799.pAlanA posanA
800.pAlana poSaNa
801.pAlana poSaNa karanA
802.pAlayitA
803.pAlita
804.pAlita poSita
805.pAmara
806.paMkhI
807.paMtha
808.pAnI
809.pApa
810.pApAcArI
811.pApahIna
812.pApahInatA
813.pApAtmA
814.pApI
815.pApiSTha
816.parabata
817.paraheja
818.parahezadAra
819.parahezI
820.paravara
821.paravarisha
822.paridhAna
823.pariMdA
824.paripAlaka
825.paripAlana
826.paripAlana karanA
827.paripAlita
828.paripoSaka
829.paripoSaNa
830.paripoSanA karanA
831.paripoSita
832.paripUrNatA
833.parivahana
834.pArivArika
835.pArivAriya
836.paRosa
837.parvata
838.parvatavAsI
839.paryApta
840.pAsa paRosa
841.pashu
842.patabhraSTatA
843.pAtakatA
844.pAtakI
845.paTApaTa
846.patha
847.pathabhraSTa
848.patitA
849.patita
850.pAva
851.pAvana
852.pAvanatA
853.pavitra
854.pavitratA

855.phajIhata
856.phana
857.pharrATe se
858.phasala kaTAI
859.phaTAphaTa
860.phaurana
861.phaurana se
862.phiranA
863.phurtI se
864.poSaka
865.posanA
866.poSaNa
867.poSayitA
868.poshAka
869.poSita
870.posta
871.prakAma
872.pralaya
873.prArthanA
874.prasAda
875.pratipAlaka
876.pravacana sabhA
877.prema
878.premapUrNa
879.prithvI
880.prIti
881.priyatA
882.pU~jI
883.pUjA
884.pUjA samAroha
885.pUjana
886.pUjArpaNa
887.pula
888.puliyA
889.punIta
890.punItatA
891.pUrA
892.pUrApana
893.pUrNa
894.pUrNAnshatah
895.pUrNAnshikatA
896.pUrNatA
897.pUrNatah
898.pUrNatayA
899.pUta
900.putra
901.putra putrI
902.putra ratna
903.putrahIna
904.putrI
905.pyAra
906.rAha
907.rAhaguzara
908.rahana
909.rahane vAlA
910.rahasya
911.rahavAla
912.rahavAsI
913.rahivAsa
914.raIsa loga
915.raIsI
916.raiyata
917.raMga DhaMga
918.rAstA
919.re~ganA
920.ripu
921.riputA
922.rishi
923.roka
924.rupayA paisA
925.sA~cA
926.sabhI
927.sabhya
928.sabhyatA
929.sabzI
930.saca
931.sacamuca
932.saccA
933.saccApana
934.saccaritra
935.saccaritrA
936.saccaritratA
937.sadAcAra
938.sadAcaraNa
939.sadAcArI
940.sadAcAriNI
941.sadAcAritA
942.sadAcaritA
943.sadAtmA
944.sadguNa
945.sadguNI
946.sAdhana sampanna
947.sAdhu
948.sAdhu mahAtmA
949.sAdhu saMta
950.sAdhutA
951.sAdika
952.sagA
953.sAga
954.sAga bhAjI
955.sAga sabzI
956.sAgapAta
957.sahI
958.sahI taura para
959.sahodara
960.saiyamashIla
961.sAja sambhAla
962.sajjana
963.sajjanatA
964.sakala
965.sakalatA
966.sakhila
967.samagratA
968.samagratah
969.samaitrI
970.samajha
971.samajhadAra
972.samajhadArI
973.samApta honA
974.samApti
975.samastatA
976.saMbhASI
977.saMdeha
978.saMdehapUrvaka
979.saMdehashIla
980.saMdehI
981.saMdigdhatah
982.saMga
983.saMgata
984.saMgati
985.saMgAtI
986.saMgI
987.saMgI sAthI
988.sAMgopAngatA
989.saMhAra
990.saMjnA
991.saMjnAna
992.saMkhyA
993.saMkitatah
994.saMmArga
995.sampadA
996.sampanna
997.sampatti
998.sampattishAlI
999.sampUrNa
1000.sampUrNatA
1001.sampUrNataH

1002.sampUrNatayA
1003.samriddha
1004.samriddhatA
1005.samriddhishAlI
1006.saMSaya
1007.saMSayAlu
1008.saMSayI
1009.saMstuti
1010.saMta
1011.saMta mahAtmA
1012.saMtAna
1013.saMtAnahIna
1014.saMtAnahInatA
1015.saMtatihInatA
1016.samUcApana
1017.samucita
1018.samUhatA
1019.saMvedanA
1020.saMyama
1021.saMyamI
1022.saMyamita
1023.saMyAsI
1024.saMyata
1025.saMyata
1026.saMyatAtmA
1027.saniSTa
1028.sannivAsa
1029.sannivAsI
1030.saphAyA
1031.saprema
1032.saRaka
1033.saRAka se
1034.sarakanA
1035.sarAsara
1036.sarovara
1037.sarrATe se
1038.sArva
1039.sarva
1040.sarva sampanna
1041.sarvAgatah
1042.sArvajanika mArga
1043.sarvaMAsha
1044.sarvAMgatA
1045.sarVAMgIyatA
1046.sarvAMshikatA
1047.sarvashah
1048.sarvatA
1049.sarvatah
1050.sarvathA
1051.sasaMdeha
1052.sasaMshaya
1053.sAshcarya
1054.sashesha
1055.saTAsaTa
1056.satkalA
1057.satmArga
1058.satpatha
1059.Satru
1060.SatrutA
1061.satsamAgama
1062.satsanga
1063.sAtvika
1064.satya
1065.satya premI
1066.satyAdhArita
1067.satyAnAsha
1068.satyaniSTha
1069.satyAnurAgI
1070.satyapriya
1071.satyashIla
1072.satyatA
1073.satyatah
1074.satyatApUrNa
1075.satyAtmA
1076.satyAtmaka
1077.satyavAna
1078.satyavrata
1079.saudAgara
1080.saudAgarI
1081.savismaya
1082.seTha loga
1083.seTha mahAjana
1084.seTha sAhUkAra
1085.setu
1086.setu baMdha
1087.shA~sata
1088.shaila
1089.shaitAnI
1090.shaka
1091.shAka
1092.shakkI
1093.shAktihInatA
1094.shAmata
1095.shaMkA
1096.shaMkAlu
1097.shaMkApUrvaka
1098.shaMkASIla
1099.shaMkI
1100.shApa
1101.shApa denA
1102.shapagrasta
1103.shapita
1104.shapta
1105.sharApanA
1106.sharAphata
1107.sharIpha
1108.sharIra
1109.shAsita
1110.shata prati shata
1111.shaThatA
1112.shIghra
1113.shIghratA se
1114.shIghratah
1115.shIla
1116.shIlA
1117.shIla
1118.shIlAcAra
1119.shIlahIna
1120.shIlavAna
1121.shIlavanta
1122.shilavantI
1123.shilavatI
1124.shishTa
1125.shiSTa
1126.shiSTa AcaraNa
1127.shiSTAcAra
1128.shiSTAcArI
1129.shiSTatA
1130.shriMga
1131.shubahA
1132.shubhAshirvAda
1133.shucitA
1134.shUnya
1135.sneha
1136.sohabata
1137.SraddhA
1138.SraddhA bhakti
1139.SrApa
1140.SreshTha
1141.SreshThatA
1142.SrimaMta
1143.SrimaMta varga
1144.stuti
1145.SubhAshisha
1146.sujana
1147.sujanatA
1148.sukarmI

1149.sukhI
1150.sumArga
1151.sumArgI
1152.supaMtha
1153.supatha
1154.suraMga
1155.suraMgA
1156.suraMga mArga
1157.sushIlA
1158.sushIla
1159.sutA
1160.suta
1161.suvittatA
1162.svabhAva
1163.svAcaraNa
1164.svAnushasana
1165.svAnushAsita
1166.svargagati
1167.svargavAsa
1168.tAbaRatoRa
1169.TahalanA
1170.tAjjuba
1171.tAlAba
1172.tana
1173.tangadastI
1174.tangI
1175.taRAka se
1176.tarakArI
1177.taruNa
1178.taruNAI
1179.tathyatah
1180.tatkAla
1181.tatkAlatah
1182.tatkSaNa
1183.tattvatah
1184.taula
1185.taura tarikA
1186.tavAhI
1187.tejI se
1188.thoRA sA
1189.tIrthayAtrI
1190.ToTA
1191.trailokya
1192.tribhuvana
1193.tribhuvanIya
1194.triloka
1195.trilokIya
1196.tuMga
1197.turaMta
1198.uccavittatA
1199.uDDayana
1200.upabhASA
1201.upadesha sabhA
1202.upavAsa
1203.upayukta
1204.uttamatA
1205.uttansa
1206.vaira
1207.vairI
1208.vAkaI
1209.vaMdanA
1210.vaMsha
1211.vana
1212.vanacara
1213.vanavAsI
1214.vandhyatA
1215.vanecara
1216.vaNika
1217.vanIya
1218.vanshahIna
1219.vanshahInatA
1220.vAnya
1221.vanya
1222.varSA
1223.varSAta
1224.vAsa
1225.vasana
1226.vAshidangI
1227.vAsI
1228.vAsitA
1229.vAsIyatA
1230.vAstava me~
1231.vAstavika
1232.vAstavikatA
1233.vastra
1234.vastutA
1235.vastutah
1236.vibodha
1237.vicalita honA
1238.vicitra
1239.vidhvansa
1240.vidveSI
1241.vihaga
1242.vihaMga
1243.vilopa
1244.vilupta
1245.vimAna cAlana
1246.vimAna sancAlana
1247.vimAnana
1248.vimArga
1249.vimArgI
1250.vimURhatA
1251.vinASa
1252.vipina
1253.vishva
1254.vishvAsa
1255.vismaya
1256.vismayajanaka
1257.vismayakara
1258.vismayakArI
1259.vismriti
1260.vrata
1261.vratopavAsa
1262.vriddhatA
1263.vriddhAvasthA
1264.vriSTi
1265.vyabhicArI
1266.vyabhicAriNI
1267.vyApAra
1268.vyApArI
1269.vyavahAra
1270.vyavasAya
1271.vyavasAyI
1272.vyoma
1273.yamita
1274.yAra
1275.yArAnA
1276.yArI
1277.yArI
1278.yAtAyAta
1279.yathArtha
1280.yathArthatA
1281.yathArthatah
1282.yathAyatha
1283.yauvana
1284.yauvana-josha
1285.yogI
1286.yuvA
1287.yuvAvasthA
1288.zabAna
1289.zara zamIna
1290.zevara
1291.zillata