

A Unified Model for Concept Structuring

THESIS

Submitted in partial fulfilment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

By

Alka Shahpoor Irani
(nee: Alka Waman Narwekar)
National Centre for Software Technology
Gulmohar Cross Road no.9, Juhu, Bombay 400 049, INDIA

Under the Supervision

of
P Sadanandan

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN) INDIA

1995

*To my family
and to the memory of my father*

Acknowledgement

Work of this kind is never complete. There are always gaps, future plans to be incorporated and newer ideas to play with. Thanks the professors, deans, and other faculty members of BITS for helping me crystalize my work. I thank all of them, especially Dr Maheshwari, Dr Saha and Dr Ganeshan for their concern, sincerity and help.

This work would not have seen the light of the day without the help of my guide Mr. P. Sadanandan. I am grateful to him for accepting me as his student and for correcting the drafts, making suggestions, evaluating the work and advising me from time to time.

For me, NCST has been like a home away from home. I am thankful to the institute for allowing me to carry on this research. I am thankful to Dr. S. Ramani, Director, NCST for giving me full intellectual liberty to pursue my studies.

I am also thankful to members of Graphics Division, especially, Dr. S. P. Mudur, Head of the Graphics group for making available many tools and interfaces for our "experiments".

There are many persons who have helped me in innumerable ways. I was fortunate to have parents who had great respect for education and higher studies. My early childhood was full of fun and people: two younger brother and a sister at home and many uncles, aunts, cousins, and friends to interact with! I am thankful to them for the formation of my "concept base".

Over the years, I came across many close associates; first, in Tata Institute of Fundamental Research and then at NCST; the software group: T. M. Vijayaraman, Sandhya Desai, V. Kamala and late Mr. V. S. Rao; the research group: Kamal Lodhaya, Ramanujam, S. Arunkumar, Paritosh Pandya, Pijush Ghosh, R Chandrasekhar; and other colleagues, to name a few! I thank them all for the fruitful interactions I had with them.

I thank M Srikant and Anurag Bhatanagar for good discussions on Ontology and NLP related work; Jitendra Loyal and Sanjay Pathak for work on the transliteration project: Rupanthar; and Ajay Gupta on OCR project.

Thanks to NCST's system support group: Bharat Desai, D. S. Rane, P. S. Khandge, S. B. Patankar and J. D. Deshmukh (the last two have left NCST).

I wish I could include the list of all the administrative staff members from various departments of NCST: Library, Accounts, Purchase, Stores, Canteen, Personnel, Transport for their excellent supporting services. Thanks specially to George Arakal, S. H. K. Iyer et al. (Special thanks to Sabrina D'Souza for assistance in typing lecture notes and to Sakpal for xeroxing the thesis.)

On all walks of life, the friends with whom I interacted with enriched my views on many subjects. Thanks to Geeta, Sujatha, Vidya, Vijju, Savitri, Sridhar, Pijush, Ramesh, Sylvia, Christine..... Thanks to Sujatha Rao and Geeta Oommen also for their assistance in making me cope with all the crises during these last three crucial years.

I am indebted to the writers who influenced me greatly: McCarthy, Newell, Simon, Hayes, Pylyshyn, Fodor, Putnam, Woods....

I really don't know how to thank Professor Narasimhan. Right from the time I first met him, I was impressed by his personality, his intellectual pursuits, his language, and his concern for the global issues. I thank him as well as T. M. Vijayaraman for their earlier comments on the presentation.

I am indebted to my family members: my husband, Shahpoor, who being a perfectionist, is more of a critic than an admirer and thereby responsible for many positive changes in my life; my in-laws, Gool and Jehangir Irani, Aunt Dolly Mehta and Aunt Banoo Boman-Behram, for giving me freedom from the domestic duties to follow this path; and my children, Tania and Porus, who provided enough evidences of child-learning to shape my ideas.

To make this thanks-giving exhaustive, I thank all those gurus, colleagues, friends, relatives and students who helped me in innumerable ways.

Finally, I thank the super-powers for their supreme design: a human being to take the lessons from!

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

PILANI RAJASTHAN

CERTIFICATE

This is to certify that the thesis entitled **A Unified Model for Concept Structuring** and submitted by **Alka Shahpoor Irani** (nee Alka Waman Narwekar) ID. No. 88PHXF801 for award of **Ph. D. Degree of the Institute**, embodies original work done by her under my supervision.

**Signature in full of
the Supervisor**



**Name in capital block
letters** P. SADANANDAN

Date: Designation

**Associate Director
NCST**

List of Abbreviations

AI	Artificial Intelligence
CIBA	Conceptual Information BAse (Acronym)
DBMS	Database Management System
DP	Deterministic Parsing
FOPC	First Order Predicate Calculus
LDOCE	Longman's Dictionary of Contemporary English
MRD	Machine Readable Dictionary
NL	Natural Language
OCR	Optical Character Recognition
Od	Object (indirect)
Oi	Object (direct)
Singlish	Streamlined English
VP	Verb Phrase

Ph.D. Thesis on
A Unified Model for Concept Structuring
by
Alka S. Irani

Language & Style corrections made in the Thesis

Chapter 1: Introduction

Page	Line	Original Text	Modified to
10	16	knowledge that is used in the furtherance of goals[Pylyshyn, 1981].	knowledge that is used in the furtherance of goals[Pylyshyn, 1981].
11	19	and knowledge retrieval - should work well individually but they should	and knowledge retrieval - work well individually but they should
11	27	What is Knowledge Representation	What is Knowledge Representation?
12	3	characterized the crux of the representation problem.	characterized, the crux of the representation problem.
12	13	The realization that most of the nature's solutions are best, made	The realization that most of the nature's solutions are the best, made
12	25	Here we summarise that work.	Here we summarize that work.
13	14	analysing the arrangement of alphabets in it.	analyzing the arrangement of alphabets in it.
15	15	We describe Conceptual Information Base CIBA and its dimensions in Figure 6.1.	In Chapter 6, we describe Conceptual Information Base CIBA and its dimensions.
15	19	(For list of some of the basic concepts see Appendix V.)	(For list of some of the basic concepts see Appendix IV.)
17	12	Our interpreter tries to give a unique sense, viz. "interpretation" to a Singlish sentence by converting it into a description (See Appendix XV).	Our interpreter tries to give a unique sense, viz. "semantic interpretation" to a Singlish sentence by converting it into a description
18	3	For the details about the interpreter refer to Documentation on CIBA interpreter -Interpret [Irani, 1995].	We have implemented a parser based on the hypothesis "deterministic parsing" suggested by Marcus[Marcus, 1980].
18	16	Intensions of the discourse structures are captured by their	Intentions of the discourse structures are captured by their
21	3	of means depending upon what users finds convenient at that point of	of means depending upon what user finds convenient at that point of

Chapter 2: Biological Information Systems

Page	Line	Original Text	Modified to
27	7	Possessive? (corresponding to has-a) in computer	Possessive (corresponding) to in computer
27	11	This, he realizes, will be possible only if he follows a mutually known	This, he realizes, will be possible only if he follows mutually known
28	3	subsection Artifacts	subsection (Artefacts)
28	4	(artifacts) to serve some purpose.	(artefacts) to serve some purpose.
32	13	as a resource to be used in the development of knowledge about the subject.	as a resource to be used in the development of knowledge about the subject.)
34	4	processing units to execute an operation.	processing unit to execute an operation.
36	11	Cognitive science rests on the foundational assumption that there exist a natural set	Cognitive science rests on the foundational assumption that there exists a natural set
37	5	In the next chapter, we will talk about entities in the computer	In the next chapter, we will talk about entities in the computer
37	6	the knowledge of the real world.	the knowledge of the real world.

Chapter 3: Computer Information Systems

Page	Line	Original Text	Modified to
39	9	domains: databases, graphics,	domains): databases, graphics.
40	4	by advocating division of tasks into subtasks.	advocating division of tasks into subtasks,
40	28	explaining system,	explaining the system),
41	8	They use inference mechanism to get implicit knowledge	They support inference mechanism to get implicit knowledge
41	12	Database technology emerged to manage, access, large amount of shared information	Database technology emerged to manage large amount of shared information
41	16	These languages are also usable along with generally provided with DBMS software.	These languages are generally provided with DBMS software.
42	5	Menu Selection	Menu Selection:
42	11	Form Fill-in	Form Fill-in:
49	13	The object-oriented programming approach is a "data-oriented" approach to software design and development, where data is encapsulated in	The object-oriented programming approach provides a level of abstraction for software design and development, where data is encapsulated in

Chapter 4: Design Considerations

Page	Line	Original Text	Modified to
53	10	different level at physical level and not at the knowledge	different level, at physical level and not at the knowledge
53	19	Language was primarily developed as a communication means, but	Over the years, language developed as a communication medium, but
57	7	The problem of getting semantic interpretation is	The problem of getting semantic interpretation of a natural language utterance is
58	26	a mechanism to retrieve information through variety of means depending upon what users finds convenient at that point of	a mechanism to retrieve information through a variety of means depending upon what users find convenient at that point of
59	2	in both. Therefore a need for "Unified Model for Concept Structuring".	in both. Therefore a need for "A Unified Model for Concept Structuring".
59	17	role in the organization of the information.	role in the organization of information.
59	18	Ability to build a structure, name it and store and get it as	Ability to build a structure, name it and store, and get it as
59	26	The chunks can form an hierarchy.	The chunks can form a hierarchy.
60	19	Knowledge organization based on intentions is like	Knowledge organization based on intensions is like

Chapter 5: Singlish - Streamlined English

Page	Line	Original Text	Modified to
70	12	Refer to table 3.1 for verb fomiss.	Refer to table 5.1 for verb forms.
73	18	Exclamations are identified by the end-mark as well as staring word which usually belong to some close-class of words.	Exclamations are identified by the end-mark as well as starting word which usually belongs to some close-class of words.
74	5	A word or phrase that is used in a novel or special way (in a metaphoric sense may be) is put in quotes.	A word or phrase that is used in a novel or special way (in a metaphoric sense may be) is put in quotes

Page	Line	Original Text	Modified to
75	11	the roles, in turn, can be a multi-word phrase with one or	roles, in turn, can be a multi-word phrase with one or
75	14	beings ability to make 'sense' out of the construction.	beings' ability to make 'sense' out of the construction.
77	21	sentence(like its complexity, number of clause, number of	sentence(like its complexity, number of clauses, number of
77	24	clear-cut boundaries. In sanskrit, compound nouns are written	clear-cut boundaries. In Sanskrit, compound nouns are written
78	28	Many English verbs (phrasal verbs) consists of two parts:a 'base' verb(like bring, take, come) and another 'small word' (like up, down, off,	Many English verbs (phrasal verbs) consists of two parts:a 'base' verb (like bring, take, come) and another 'small word' (like up, down, off,
80	3	The punctuations and markers a used for streamlining are as follows:	The punctuations and markers used for streamlining are as follows:
80	5	Relative clauses are enclosed between a pair of double-backslashes.	Relative clauses are enclosed between a pair of double-backslashes.
80	9	In constructs using 'infinity to' , to is to be connected to the verb by tilde. In case of ambiguity, the head of a noun phrase can be marked	In constructs using 'infinity to' , to can be connected to the verb by tilde. In case of ambiguity, the head of a noun phrase is marked
80	13	Connections between head and modifiers can be explicitly shown by	Connections between head and modifiers are explicitly shown by
80	17	A part of speech can be enclosed in a square brackets.	A part of speech can be enclosed in square brackets.
84	6	Frame-based languages which have slots resembling the intentions	Frame-based languages which have slots resembling the intentions
84	17	Every chuld paintakingly going through remembering all the past and present	Every child going through the trauma of remembering all the past and present

Chapter 6: Conceptual Information Base CIBA'

Page	Line	Original Text	Modified to
89	16	is that of the representation schemes. Representation schemes: internal	is that of the representation schemes. Representation schemes, internal
89	17	spoken languages, written languages are built incrementally. Each one having its own plane, having	spoken languages and written languages are built incrementally, each one having its own plane, having
89	21	It follows from this that the computer plane, if it has to be truly representational, should have the mappings of concepts from all the three planes in	To make the computer plane truly representational, the mappings of concepts from all the three planes should be provided in
90	11	One of the main characteristic a representation must have is "semanticity".	One of the main characteristic, a representation must have is "semanticity".
91	13	For human beings the way to general concepts is 1) through a raw form: through specific knowledge as human beings live and come across various "experiences" in the word and they have a mechanism to "generalize" or 2) through a compiled form: through written text,	For human beings the way to general concepts is 1) through a raw form, through specific knowledge; as human beings live and come across various "experiences" in the world and they have a mechanism to "generalize" and 2) through a compiled form: through written text,
95	22	category. Section 5 explains these dimensions.	category. Section 6.11 explains these dimensions.
96	6	The intentions of these description structures are captured by the	The intentions of these description structures are captured by the
96	27	Paninian framework designed more than two million	Paninian framework designed more than two thousand
97	20	For a list of basic concepts see appendix I	For a list of basic concepts see appendix V.
98	30	All these words stand for person in PLATAEU	Figure 6.3 All these words stand for 'person' in PLATAEU [Irani, 1991]
99	17	Other operators are and or not sameAs, oneOf, someOf noneOf and	Other operators are and, or, not, sameAs, oneOf, someOf noneOf and
100	10	this Church since April?" Refer to Appendix VII for some examples.	this Church since April?" [Date, 1994]
100	27	the computational model of the world.	a computational model of the world.

Chapter 7: Text-based System for Conceptual Knowledge

Page	Line	Original Text	Modified to
104	9	In Chapter 5, we have seen how we can represent basic concepts	In Chapter 6, we have seen how we can represent basic concepts
105	23	Our work is based on the following hypotheses	Our work is based on the following hypotheses:
106	7	The books are a common carriers of human knowledge. While being written by an	The books are common carriers of human knowledge. While being written by an
106	17	The effacacy of a knowledge engineering methodology is diminished if it is not supported by software	The efficacy of a knowledge engineering methodology is diminished if it is not supported by software
106	22	Selection the best book using the selection criteria given below.	Selection of the best book using the selection criteria given below.
108	22	out the organizational information about the text(see Figure 1). accounting ionformation	out the organizational information about the text(see Figure 7.1).
109	33	Skeleton descriptors through visual clues	Figure 7.1 Information Organization
110	18	We should be able to determine the ideal(or near-ideal) way of presenting the text on a	We should be able to determine the ideal(or near-ideal) way of presenting the text on
111	17	In a physical sense, in text-books is linearly ordered.	In a physical sense, text in text-books is linearly ordered.
113	15	explosion in the bill-of-material problem).	explosion in the bill-of-materials problem).
114	3	over all b	(over all b)
116	2	Skeleton descriptions are basically titles conveying intentions of	Skeleton descriptions are basically titles conveying intensions of
116	4	In Chapter 9, we have described an interpreter which can convert	In Chapter 10, we have described an interpreter which can convert
116	12	In this paper we have suggested criteria for selection of the ideal	In this chapter, we have suggested criteria for selection of the ideal

Chapter 8: Modelling Factual Information

121	12	Thus on the one hand, there is the informal system of slot-names, procedure names (from which human	Thus on the one hand, there is the informal system of slot-names (from which human
126	29	Efforts are on to build natural language like user interfaces	Efforts are on to build natural-language-like user interfaces
128	9	where data values come from the expressions using abstract data types	where data values come from the expressions using abstract data types
129	23	Structuring of elements is not arbitrary but with some purpose(intention) in mind.	Structuring of elements is not arbitrary but with some purpose in mind.
132	3	A description has to parts: a part corresponding to things	A description has two parts: a part corresponding to things
132	8	important attributes and relationships are important to know for this	attributes and relationships important to know for this
135	3	Suppose we have the following facts about students in a college to start with.	Suppose we have these facts about a student in a college to start with.
135	20	For example: Vijay (29, 180,55) Sujatha (26,164,49) Smita (25, 150,45) Sandhya (27, 160,50) Sheela (25,163,51)	For example: Identifier age height weight Vijay (29, 180,55) Vijay 29 180 55 Sujatha (26,164,49) Sujatha 26 164 49 Smita (25, 150,45) Smita 25 150 45 Sandhya (27, 160,50) Sandhya 27 160 50 Sheela (25,163,51) Sheela 25 163 5
137	10	categorized into one of the three matabehavioural categories,	categorized into one of the three meta-behavioural categories,

Chapter 9: ReWire - Real World Information Retrieval

Page	Line	Original Text	Modified to
149	5	provide cue to further matching is what is called intention	provide cue to further matching is what is called intension
149	8	intentions of queries should be matched with intentions of	intensions of queries should be matched with intensions of
149	10	Thus it is necessary in a knowledge-based system to capture intentions	Thus it is necessary in a knowledge-based system to capture intensions
149	12	What is Intentional Organization?	What is Intensional Organization?
149	13	The concept of intention dates back to Frege[Frege, 1949] and his distinction between the	The concept of intension dates back to Frege[Frege, 1949] and his distinction between the
149	15	The concept of intention	The concept of intension
149	16	Montague defines intention of any expression as a function from a set of points	Montague defines intension of any expression as a function from a set of points
149	19	Thus the intention of a name is a function which, given any index, picks out	Thus the intension of a name is a function which, given any index, picks out
149	21	Similarly the intention of a set picks out some collection of individuals which is the referent of the set-name at each index, and the intention of a	Similarly the intension of a set picks out some collection of individuals which is the referent of the set-name at each index, and the intension of a
150	9	The intentions of data structures should be formed out of the	The intensions of data structures should be formed out of the
152	3	For example, intention of an hierarchy may be to capture partonomic	For example, intension of an hierarchy may be to capture partonomic
152	6	Once structures and their intentions are known,	Once structures and their intensions are known,
152	28	We will now worry about how to get intentions of descriptions as well as intentions of structures i.e. meta-knowledge when our	We will now worry about how to get intensions of descriptions as well as intensions of structures i.e. meta-knowledge when our
153	1	How to Get Intentions of Descriptions	How to Get Intensions of Descriptions

Page	Line	Original Text	Modified to
153	3	In text, titles and sub-titles capture the intentions of the text that	In text, titles and sub-titles capture the intensions of the text that
153	19	theme) of the sentence and (2) capturing the intention (The intention of a sentence may in turn depend upon the kind of the	theme) of the sentence and (2) capturing the intension (The intension of a sentence may in turn depend upon the kind of the
153	23	It was summer (intention:specifies time) a crow was flying to go somewhere(intention:specifies Action) the crow was very thirsty (intention:specifies State) his threal was dry with thirst (intention:specifies characteristic) he was suffering a lot(intention:specifies state) he was looking in all the directions(intention:specifies act) but there was no water(intention:specifies state) he was tired(intention:specifies state) he perched on the branch of a tree(intention:specifies act) he was now thinking(intention:specifies mental act)	It was summer (intension:specifies time) a crow was flying to go somewhere(intension:specifies Action) the crow was very thirsty (intension:specifies State) his throat was dry with thirst (intension:specifies characteristic) he was suffering a lot(intension:specifies state) he was looking in all the directions(intension:specifies act) but there was no water(intension:specifies state) he was tired(intension:specifies state) he perched on the branch of a tree(intension:specifies act) he was now thinking(intension:specifies mental act)
154	3	The question now is, how to get these intentions from the text. Here we	The question now is, how to get these intensions from the text. Here we
155	2	For fact-based systems, intentions of the structures can be	For fact-based systems, intensions of the structures can be
155	4	intentions can be captured from data descriptors, if	intensions can be captured from data descriptors, if
155	10	Intentions of Input Descriptors	Intensions of Input Descriptors
155	12	that is wanted(intention), while the rest of the description describes what is the	that is wanted (intension), while the rest of the description describes what is the
155	17	'from' specifies , the class(relation) of interest;	'from' specifies , the class (relation) of interest;
157	12	What forms the intention(data descriptor) and what forms an	What forms the intension(data descriptor) and what forms an
157	17	the intention should provide a 'handle' to lift the data. In other words, intention and extension should correspond to the title and body of a chunk	the intension should provide a 'handle' to lift the data. In other words, intension and extension should correspond to the title and body of a chunk

Chapter 10: Implementation

Page	Line	Original Text	Modified to
158	15	concept mappings and concept clusters in Appendix XI.	concept mappings and concept clusters in Appendix IV, VII and XI respectively
160	4	The overall algorithm for interpreting a Singlish(Streamlined or even simple English)	The overall algorithm for interpreting a Singlish(Streamlined)
161	13	For instance, they do not normally throw the semantic information away.	For instance, they do not normally throw the semantic information away
168	32	using our parallel algorithm.	using our parallel algorithm
169	3	of the Topic and Intension	of the Topic and Intension
169	12	intention and extension in formal semantic.	intension and extension in formal semantic.
169	17	in to consideration what is called Language in use[Narasimhan,	into consideration what is called Language in use[Narasimhan,
169	24	Let us illustrate this with some example.	Let us illustrate this with some examples

Chapter 11: Conclusion

Page	Line	Original Text	Modified to
185	17	We will feel institutionalizing Singlish language(or something similar)	We feel institutionalizing language like Singlish

Appendix 8

It was in a nested-lists format in the original manuscript, now it is changed to tree-like format.

THE NEW THESIS DOES NOT DIFFER FROM THE ORIGINAL EXCEPT FOR THE CORRECTIONS LISTED IN PAGE NO. 1 - 10.

(P. Sadanandan)
January 2, 1995

Contents

1	Introduction	2
1.1	The Problem	2
1.2	Our View	5
1.2.1	Biological Information Systems	6
1.2.2	Singlish - Streamlined English	8
1.2.3	The Interpreter	9
1.2.4	Skeletonizer	10
1.2.5	Streamliner	11
1.2.6	Acquisition of Simple Concepts	11
1.2.7	Acquisition of Complex Concepts (Terminology):	11
1.2.8	ReWIRe - Real World Information Retrieval	12
1.3	Importance of an Interpretational Base - CIBA	13
2	Biological Information Systems	15
2.1	A View of Biological Information Systems	15
2.1.1	A Need to Communicate	17
2.1.2	Identifiers and Classes	18

2.1.3	Attributes	18
2.1.4	Relationships	18
2.1.5	Grammar	19
2.1.6	Learning to Communicate	19
2.1.7	Domains	19
2.1.8	Artifacts	20
2.2	Role of Language in providing Second Level Representation	20
2.3	Importance of the Concept Base	24
2.4	Importance of Discourses - Knowledge Representation with the Help of Language	24
2.4.1	Knowledge Structures and Incremental Knowledge	25
2.5	Characteristics of Organization of Knowledge by Biological Systems	26
2.6	Functionalities Expected from a Symbol System	27
3	The World of Computers: Review	30
3.1	A Brief History of Progress	30
3.2	Computer Languages	31
3.2.1	Representation languages	33
3.2.2	Data Manipulation Languages	33
3.2.3	Query Languages	34
3.3	Theoretical aspects	35
3.3.1	Computability	35
3.3.2	Relational Model	35
3.3.3	A Uniform Technical Framework - LISP	36

3.3.4	Resolution Principle	37
3.3.5	The Knowledge Level Approach	37
3.4	Knowledge Representation and Computer World	38
3.5	Formalisms for Representing Knowledge	38
3.5.1	Production System	38
3.5.2	Schema-based Formalisms	39
3.5.3	Semantic Networks	40
3.5.4	FOPC (First Order Predicate Calculus)based Sys- tems	40
3.5.5	The Object-oriented Environment	41
3.5.6	Connectionist Formalism	41
3.6	Conclusion	42
4	Design Considerations	43
4.1	Introduction	43
4.2	What can we inherit from Biological Information Systems	44
4.3	Role of Language	45
4.4	Problems with a Natural Language as a Representation Language	46
4.5	Alternative - Formal Languages	46
4.6	Why Semantics?	46
4.7	Natural Language as a Universal and Stable System for Representing Knowledge	47
4.8	Why Formalization is Difficult	48
4.9	What Needs to be Done?	49

4.10	Guiding Principles	51
5	Singlish - Streamlined English	53
5.1	Introduction	53
5.2	Overview of Basic English	54
5.2.1	Functions of Major Types of Words	55
5.2.2	Simple Sentences and Syntactic Roles	55
5.2.3	Mappings from One Category to Another: Verbals	57
5.2.4	Phrases	58
5.2.5	Semantic Roles of Various Syntactic Roles	59
5.2.6	Co-ordination	61
5.2.7	Complex Sentence	61
5.2.8	Relationship between Category and Functional Role	62
5.2.9	English and Syntactic Devices to Help in Understanding the Meaning	64
5.3	Problems in Parsing English	66
5.4	Necessity for Streamlining English	70
5.5	Streamlining English	71
5.5.1	Punctuations and Markers	72
5.5.2	Inflections	72
5.5.3	Examples of Singlish Text	73
5.6	Comparison with Other Methods	76
5.7	Advantages and Future Work	77
6	Conceptual Information Base 'CIBA'	78

6.1	Introduction	78
6.1.1	An Ideal	79
6.2	What is Meant by a Knowledge-based System	79
6.3	The Need for a Concept Base	79
6.3.1	Mental Plane of Internal Language	79
6.3.2	Discourse Plane of Spoken Languages	80
6.3.3	Discourse Plane of Written Languages	81
6.3.4	Computer Plane	81
6.4	When is a Representation Truly Representational	82
6.5	Issues in Representation of Conceptual Knowledge	82
6.6	Conceptualization - a Mechanism to Optimize Information Storage and Retrieval	84
6.7	Need to Build a Concept Base	85
6.8	The Problem	85
6.8.1	Problems with Machine Readable Dictionaries	85
6.9	How to Capture the Unique Senses	87
6.10	Nature of the Concept Base 'CIBA'	87
6.11	Methodology	89
6.11.1	Basic Concepts	89
6.11.2	Primitives	89
6.11.3	Concept-Operators	91
6.11.4	Mappings	92
6.12	Concepts in the Computer Plane	92
7	Text-based System for Conceptual Knowledge	96

7.1	Introduction	96
7.2	Our Strategy	97
7.2.1	Conversion of Books into Machine Readable Form	99
7.2.2	Getting the Logical Structure of the Body of the Book	100
7.2.3	Criteria for Selection of a Text-book on a Subject	102
7.2.4	Invariance of the Subject	102
7.2.5	Formation of Skeleton for the Knowledge Base	107
7.2.6	Transformation of Skeleton Descriptors	108
7.3	Conclusion	108
8	Modelling Factual Information	109
8.1	Introduction	109
8.2	The Problem	111
8.3	The Scenario	114
8.3.1	Factors Influencing Information Structuring	115
8.4	The Problem with Existing Formalisms	118
8.5	Our Approach	120
8.5.1	Advantages of a Concept Base Underlying Descriptions	121
8.5.2	Need for Structuring	121
8.5.3	Need for Clustering	123
8.5.4	Need for a Framework	123
8.5.5	Data Structuring Viewed as a Compression Technique	123

8.5.6	Example of Compressions in a Fact-base	127
8.5.7	Associations between Discourse Structures and Data Structures in Computers	128
8.6	Organization and Scope of Queries	128
8.7	Conclusion	133
9	ReWire - Real World Information Retrieval	134
9.1	Introduction	134
9.2	Retrieval in Computer Systems	136
9.3	Retrieval in Biological Information Systems	137
9.3.1	Kind of Information Based on Formal Appearance:	137
9.3.2	The kinds of information based on usage	137
9.4	Problem Search and Knowledge Search	138
9.5	Competence	140
9.6	What is Intentional Organization?	141
9.7	Performance	142
9.7.1	Levelling	142
9.7.2	Partitioning	143
9.7.3	Pointing	143
9.7.4	Clustering	143
9.8	Meta-knowledge	143
9.9	How to Get Intentions of Descriptions	145
9.9.1	Text-based Systems	145
9.9.2	Fact-based Systems	147
9.10	Intentions of Input Descriptors	147

9.11 Retrieval through Inferences	147
9.12 Conclusion	149
10 Implementation	150
10.1 Introduction	150
10.2 The Interpreter	151
10.2.1 The Algorithm	152
10.2.2 The Parser	154
10.2.3 Our Approach	159
10.3 From Syntactic Roles to Semantic Role: Identification of the Topic and Intention	161
10.3.1 Viewing Knowledge at Various Levels	162
10.4 Skeletonization	163
10.5 Mappings between Structures in Two Planes	164
10.6 The Meta-language for Writing Fact-base	164
10.7 Future work	165
11 Conclusion	166
11.1 Summary	166
11.2 Evaluation of CIBA	169
11.2.1 Unique Aspects	169
11.2.2 CIBA and Semanticity	169
11.2.3 Justification for Our Approach	171
11.3 Related Work	171
11.3.1 Japanese Real World Computing Program	173

11.4 Contributions of this Research 175
11.5 Shortfalls 176
11.6 Future Work 177

Preface

I can hardly believe that I could put all that I had to say on the paper in concrete form. This work towards Ph.D. is spread across 10 semesters in the form of study reports apparently having no connections with each other, and it appeared to be a gigantic task to reconcile it into a coherent thesis. Well, it is done! And now when I reflect on that, I realize that it was all towards one goal: the goal of making knowledge-driven systems 'work'!

Before commenting on the work, let me describe the background.

When I started my research in search of "A Unified Model for Concept Structuring", two questions were bugging me. First one was "How to structure the information which is not stereotype, factual, repetitive data but conceptual in nature" and the second one, "If one finds a way to structure particular information (concepts) in a particular way, what is the justification for doing so" I was looking at many computer and non-computer systems to help me arrive at the answer. I became very skeptical about the over-ambitious computers systems designed to *understand* stories, motives plans and so on. The methodologies were, perhaps good to solve "toy" problems. I felt scaling up is a major issue and even the simpler problems are not simple.

I started my search for a "solution" with fundamental questions: What is information? What is knowledge? What are computer systems trying to do? Where can we get a model that will help us build information systems?

Nature always fascinated me. And I believed in Einstein who said "Natures solutions are simple and elegant".

In my pursuit, I found an answer to another question which was troubling me over the years, and the two together solved the problem.

The question was: what is "it" that separates living things from non-living? The distinctions between living and non-living things such as

'living beings respond to the environment', 'living beings reproduce' etc. are alright as characteristics but there must be something more fundamental than that.

And I think at last I found the answer!

Human beings are biological systems. They are living systems. One important and fundamental difference between living and non-living systems is that they are 'programmed' by nature (or god or evolutionary forces or whosoever you may think of) to have wants, wishes, fears, senses to grasp various aspects of situations, and act accordingly. They can assimilate information, represent it internally and act on the basis of these representations. Living beings are information-driven (or knowledge-driven) systems.

Among living things, human being is a special species. I am convinced that human beings are special because they evolved culturally and it was the 'language' which made their progress possible by providing a way for externalization of knowledge.

Computers have provided another powerful tool to mankind; to mechanically solve problems, to store data, to communicate messages, and to do a whole lot of other things. Yet, a lot is needed if computers are to be used for knowledge representation in the true sense.

One important characterization, we have attempted to identify in this thesis is that of the representation systems. Representation systems: internal language of mental representations, spoken languages, written languages are built incrementally. Each one having its own *plane*, having what we call a "concept base". Each succeeding plane, provides a mappings for *concepts* from the earlier plane; in addition to that, it has its own "vernacular" concepts. It follows from this that if computer systems are to be used for knowledge representation, the computer plane should also be incrementally built. In our thesis, we have described a methodology to build the concept base for the *computer plane* that can be used as a *knowledge representation scheme* by knowledge-driven systems.

We have proposed natural languages not only as a medium of interaction, but also as a medium of knowledge representation in computers. We firmly believe that behind the apparent idiosyncrasies of natural languages must be a very stable system (otherwise they would have lost their functionality). We have also proposed a methodology to “streamline” a natural language.

Among various advantages of representations using a natural language, one that I would like to emphasize here is the advantage we get of having a wealth of knowledge already encoded in the scheme in the form of printed text compared to computers. If computer systems can inherit all that knowledge from books it will be a great leap. Moving from paper to electronic medium offers unprecedented potential for structuring of textual and graphical information.

I strongly believe that canonical and unambiguous representation of knowledge with flexible input-output gateways is crucial for the world that hosts as many as 2500 languages.

Unfortunately, here in this world, there are too many natural languages and most of the knowledge is in English. What about people who don't know English? Should they be deprived of the factual and conceptual knowledge that is expressed in English?

A partial solution to this problem is to translate the work from English. But there seems to be a problem.

It is remarkable, that natural languages used by human beings at places far away from each other have remarkable similarities. Most of them have the functionalities that make translation possible. The problem is that they lack vocabulary. The generation of new words and socialization of the words have to keep pace with the progress of mankind. An alternative that can be thought of is “transliteration”: picking up of words from another language and blending them into your language. This will certainly help in unifying the next-generation terminology across languages.

There can be many applications built on the top of this concept base for

the computer plane (We call it CIBA); tutoring systems, machine translation systems, information systems, user-friendly interfaces, factual information bases; etc.

Though we have enough implementation work to present, we haven't recorded much in this thesis. We have also not presented and discussed related work in details due to space and time restrictions.

We feel that the main thrust of the thesis is in ideas and in modelling.

Before I finish, let me have a confession! I had proposed the title "A Unified Model for Concept Structuring" for my thesis way back in 1989. I thought I will be unifying all the structuring mechanisms in the *computer plane*. Now after finishing the thesis, I realized that it is the *scheme for representation of knowledge on the medium which is a computer* that is getting unified with the earlier representation schemes of human beings. However, the title, the natural language description, still makes sense in general!

Chapter 1

Introduction

1.1 The Problem

Technology, by which I mean to include not only the design of artifacts, but everything that involves a codified system of methods or techniques provides both instrumental tools to help us make observations and calculations, and conceptual tools that help us to see things in new ways. Such conceptual tools may be thought of as imagination prosthetics, because they typically extend the range of the conceivable. Conceptual tools dominate periods of intellectual progress.

— [Pylyshyn, 1981]

What is a knowledge-driven system

The behaviour of intelligent organisms[man or machine] is typically explainable only if we assume that their actions are governed by decisions based on knowledge that is used in the furtherance of goals[Pylyshyn, 1981]. When the system is a machine whose behaviour is governed by what it represents - what it “knows” - it is called knowledge-driven system. The objective of our work is to provide a *Conceptual Basis* for knowledge-driven systems.

There are many ways known of systematically representing different kinds of knowledge in a sufficiently precise notation that it can be used in, or by, a computer program to solve problems. However, when the knowledge is vast and complex, the computer systems using this knowledge are likely to collapse under their own weight. Though the advances in computer hardware have made it possible to have tremendous processing power and very large capacity to store the information, the basic issue of how to model the complex data has remained a challenge.

Need for people and machine to understand each other

In his 1971 Turing award lecture [McCarthy, 1971], McCarthy emphasized generality as an essential characteristic of computer systems. Generality implies representing knowledge of various sorts and retrieving it in a way that can be useful to people. It is not easy to make a system general-purpose. The construction of large, knowledge-based applications is a complex task that comprises a number of activities and involves various participants. Not only should the components - knowledge acquisition, knowledge representation and knowledge retrieval - work well individually but they should work in co-operation with one another and also with the humans who are the creators, mentors and beneficiaries of these systems. To meet this requirement, it is mandatory that both the system and people understand each other's language.

Canonical and unambiguous representation of knowledge with flexible input-output gateways is crucial for the world that hosts as many as 2500 languages.

What is Knowledge Representation?

Knowledge representation is a medium of human expression, in which we primarily describe various aspects of the world. Any representation is fundamentally a surrogate [Davis et al, 1993]. It is not a thing in itself. Thus a knowledge representation formalism can be judged for its adequacy by finding how it maps the two worlds (external world) and (internal world or mental world) of human beings. The convention it uses for this purpose must be universal and stable.

Simultaneously producing good candidates for each of the three ingredients - the representation language, the inference regime and the particular domain knowledge - is as David Israel [Israel, 1983] characterized, the crux of the representation problem.

The Scenario

Earlier work in Artificial Intelligence (carried on almost for two decades) made it clear to researchers that knowledge is power.

Thus knowledge representation, acquisition and retrieval drew considerable attention. The three cannot be separated as they are closely linked. The way knowledge is acquired affects its representation and knowledge representation ultimately decides what is retrievable in real time.

The realization that most of the nature's solutions are the best, made people look into the nature's knowledge representation systems and cognitive science came into picture.

We do not intend to survey the field in detail as many good surveys are available. *Minds, Brains and Computers, Perspectives in Cognitive Science and Artificial Intelligence* [Ralph et al, 1992] is a good collection of work that covers computational models for knowledge-driven systems. In order to make the task of surveying manageable, they have grouped various views into two broad computational paradigms-Classical symbol processing ([Pylyshyn, 1984], [Newell, 1982], [Schank, 1972] and [Windograd, 1983]) and Connectionism ([McClelland, 1992], and [Arbib, 1992]). Both paradigms try to explain how the mind works. Here we summarise that work.

One of the fundamental claims underlying the Classical paradigm is the so-called physical symbol system hypothesis, which maintains that both the brain and a (properly programmed) digital computer are examples of physical symbol systems [Newell and Simon, 1976]. They also claim that the fundamental operations of these systems, at some level of organization, involve the disciplined manipulation of symbols, where by *symbols* we mean things which stand for or refer to something else. When describing cognitive phenomena, these symbols have come to be

called *mental representations*, and within Classical theories, they are generally construed either as propositions or models.

Another paradigm that has drawn considerable attention is the Connectionist model. Connectionist systems are based on Neuron-like architecture. A simplistic view of the mechanism is as follows: A connectionist network consists of a large number of nodes and interconnections. Each node is a simple processing unit, which can save a single value, can calculate a new *activation value* based on its old value and *activation values* passed to it and can pass its *activation value* to other nodes.

We have reservations about the connectionist mechanism. While following it, we are mimicking the brain *hardware*. It is like looking at the bit maps and understanding what the program is doing. To give another analogy, it is like understanding a chapter by analysing the arrangement of alphabets in it.

Connectionist mechanism will work only if we have a theory about where the concept should be placed in the entire network. It cannot be ad-hoc or we can't expect it to work on its own. Biological systems are capable of placing concepts at relevant places and connecting them in a certain way. (How they do it is a million dollar question!) But these capabilities are not present in a mechanical computation system. (and as yet we have no theory to help us in that direction.) Only by understanding the principle, perhaps, we can make the mechanical computation system work in similar ways.

1.2 Our View

We all know that symbol-processing systems of today are far from satisfactory. Their performance doesn't match their potential.

The failure of these systems can be attributed to a certain extent to the usage of ancient methods - the methods that were used to tackle the problems of a different kind. Earlier tasks handled by computers had a formal base. Underlying formalisms were known; they were mostly

from well-disciplined domains. People explicitly knew the methods and therefore could write algorithms which could eventually be converted into programs.

Another cause of failure can be the insensitivity to the characteristics of both Biological Information systems and Computer-based systems. While mimicking the brain, in order to exhibit the flexibility with which it processes information, designers of computer systems should take into account the capabilities and functionalities of *living organisms* too. We have taken into consideration the tools developed by the living-organisms and the limitations of non-living machines like computers while designing our system.

1.2.1 Biological Information Systems

The nature of information we carry in our head is not still very clear to us. We know a lot about various “things”, about their various “aspects” in various “ways”. In the physical world, these things can roughly be classified as objects or agents or events or states and the aspects as their characteristics or attributes which have various dimensions like time, space, colour, participants etc.

We, human beings, have found means of optimizing this information internally through naming, abstractions, categorizations or clustering.

The question is: can we also follow a similar scheme in computer-based systems? If so, how can we get the abstractions, generalizations or clusters that are universal, stable and useful?

The key to abstractions or categorizations or groupings made by humans is through language as **abstractions/categories/groupings of interest are retained in the language as lexical units(words)**.

Thus a language plays a very important role in *biological information systems*. Language builds our conceptual frame of mind. Words of a language are stable, universal, meaningful and atemporal units. They represent *concepts* for the community that uses them. We will be explaining in chapter 4, how the conceptualizations provided by a

language essentially provide a framework for the language-using community for information representation, acquisition and retrieval.

Our Conceptual Information Base 'CIBA' is built taking into account these points. CIBA plays an important role in providing *the interpretational base* for knowledge-driven systems.

CIBA consists of the *concept base* which represents *concepts* and the CIBA environment: Singlish, Interpreter, Skeletonizer and Streamliner. We will describe them here briefly.

In a natural language, very often, words represent concepts. However, one word in general has many meanings. In order to capture a unique sense represented by the word in a language, we *restrict* its sense by providing other *dimensions* to it. These dimensions are *primitive concept, basic concept, plane* and *domain* and *category*. These dimensions associate various features with the concept. We feel that these *dimensions* are enough to pin down the meaning of a *concept*. We describe Conceptual Information Base CIBA and its dimensions in Figure 6.1.

Basic concepts are the concepts a normal adult is familiar with. Basic concepts are the concepts universally known and generally have surface words representing them in a language. (For list of some of the basic concepts see Appendix V.) *Primitives* classify *concepts* into partitions. The broad classes into which the primitives fall are *Act, Agent, Object, Entity, State, Happening(Event), Theme, Information, Time, Space, Property, Quality*. ****ps****(See Appendix V for the details). Our dimension *category* corresponds to the grammatical categories in a natural language grammar. (The members of this category are noun, adjective, adverb, preposition etc.) CIBA provides a mechanism to form *descriptions* using *concepts*. A *description* describes an object or an agent or an entity or a relationship which has a *referent* at one of the four *planes*: physical plane, mental plane, discourse plane or computer plane.

For example:

The rock is next to the river. (physical plane)
He is unhappy. (mental plane)

This Paragraph is short. (discourse plane)
DO WHILE (number == 100). (computer plane)

In CIBA, a *domain* corresponds to a subject. Examples of domains are mathematics, geography, etc. (See Appendix VI for the listing of *domains*.) In a natural language, there are certain transformations, like noun to verb, verb to noun, noun to modifier and adjective to adverb. Corresponding transformations in CIBA are termed as *Concept Mappings*. (See Appendix VII for the examples of *mappings*.)

1.2.2 Singlish - Streamlined English

We found, after studying English Language Grammar [Quirk et al, 1985] and a number of model sentences from English Language Usage [Swan, 1980] that *streamlining a natural language* can help us in getting a unique meaning corresponding to an ordinary grammatical sentence of a natural language.

Some of the techniques we use are:

- use of punctuations, to separate parts of speech
- restricting use of capital letter to identifiers (proper names); our sentences do not start with capital letters
- use of hyphens to form complex noun phrases
- use of brackets to make attachments explicit

Singlish is very much like English, Singlish parser was tried on English sentences too. (Appendix I gives all the sentences under consideration, Appendix II gives the illustrative subset of sentences from Appendix I which could give roles to all constituents uniquely, Appendix III gives the problem sentences.)

1.2.3 The Interpreter

We work on the assumption that there is no fundamental difference between a natural language and a formal language when it comes to parsing sentences if we can disambiguate parts-of-speech.

A natural language becomes context-sensitive because

- most of the common words it uses have many meanings.
- it is meant for human-to-human communication, therefore whenever possible, people using it take *short-cuts*.
- descriptions in natural language are some times context-based.

Interpreting sentences in natural languages also becomes problematic because we don't take into account the *language in use*.

Our interpreter tries to give a unique sense, viz. "interpretation" to a *Singlish sentence* by converting it into a *description* (See Appendix XV). Semantic Interpretation has been defined as the process of mapping a syntactically analyzed natural language text to a representation of its meaning (Hirst) [Hirst, 1987]. It has also been generally recognized that compositionality (meaning of the whole is a systematic function of the meaning of the parts) will remain a central concern of all semantic theories. An extreme position that has been taken is that language interpretation can be ultimately viewed as a process of word-sense discrimination (Rieger) [Rieger, 1976].

The problem of getting semantic interpretation is difficult, because, as we have already mentioned earlier, in a sentence, the structure is flattened. There is a mixing of boundaries. A word in a sentence can be a *head word*, which has a part to play as one of the *roles* in a sentence, or it can be a *modifier* related to a head word.

We have used *reduction method* to disambiguate parts-of-speech of a sentence.

The idea is to determine the *concepts* underlying surface words, the *roles*

of *concepts* and *identifiers* in a sentence using positional information, punctuations, closeness, compatibility, language usage and other cues.

For the details about the interpreter refer to *Documentation on CIBA interpreter -Interpret* [Irani, 1995].

1.2.4 Skeletonizer

Skeletonizer builds information structure.

Organization of *descriptions* is very important. To get the right information, this organization along with the organization criteria should be made available. The organization is a kind of compression that delimits the scope of queries.

A particular organization is selected because of the promise it holds. It is selected if it can deliver the goods; that is, if it is accessible for browsing or querying.

Organization of Textual information

A structure is called a *discourse structure* if it serves a particular purpose. Intensions of the *discourse structures* are captured by their *discourse types*.

Examples of *discourse types* are *definition, example, story, explanation, problem, question, answer, method, procedure*.

Organization of Factual information

In typical knowledge-based systems we find structures so that

- descriptions about the same object or descriptions of an instance of a state are grouped together to form records(same time)
- descriptions of instances of a class of objects to form *databases*

Factual information can be put into any of the formalisms like *database, network, frame*. We view them as compression mechanisms, and associate *descriptions* with them.

1.2.5 Streamliner

It is easier to write in Singlish. However to automatically retrieve the text for text-books, a pre-processor is needed to convert ordinary English into Singlish. This can be done by the *Streamliner*.

1.2.6 Acquisition of Simple Concepts

Simple concepts are concepts that can be interpreted in terms of *basic concepts, primitives, concept operators, concept mappings*. (Most of the words in a dictionary come under this category).

With advances in OCR technology, it is possible to get text corresponding to these concepts. Another source for simple concepts is a machine readable dictionary. The interpretations corresponding to various words are helpful in getting dimensions associated with the concepts underlying these words.

1.2.7 Acquisition of Complex Concepts (Terminology):

A complex concept corresponds to *terminology* in a specific subject or an encyclopedic entry. A *complex concept* is defined using a *chunk*. Once the initial concept-base with its *interpreter* for descriptions is ready (first order concept-base), it is used for knowledge acquisition in various domains like Mathematics, Physics, Chemistry in order to store that knowledge in a modular fashion.

Here we rely on existing sources to get knowledge as well as its organization. The knowledge structures that are directly accessible to observations and analysis and mirror biological information systems are the methods and techniques of information organization which the world of printing technology has given us. Thus we propose a methodology in Chapter 7, which uses the text-books on a subject as a resource to be used in the development of a concept-base about the subject.

It is widely accepted that the effective development of a knowledge-based system depends heavily on its knowledge elicitation also. We view knowledge engineering as a process of information transformation in which knowledge is acquired and ultimately transformed to a formal representation. We feel that the prescriptive frameworks currently outlined for knowledge engineering give little assistance on how to tackle typical knowledge acquisition and representation problem.

The methodology we follow for acquiring knowledge from text-books is as follows:

- Conversion of books into machine readable form.
- Getting the logical structure of a book (organization).
- Selection of the *best* book using the selection criteria specified in Chapter 7.
- Formation of the *skeleton* for the knowledge-base on the subject.
- Transformation of *titles* picked from the *raw text* into *descriptors*.
- Filling the *skeleton* using the original text.

The important point to note is that we are using printed text not only for knowledge contents but knowledge organization at a higher level of granularity.

We believe that our concept-base CIBA provides a conceptual tool for knowledge-driven systems.

1.2.8 ReWIRe - Real World Information Retrieval

We strongly believe that justifications for data structures are necessary. The soundness of a particular structure can be verified if only the characteristics of the data to be stored in the structures are made explicit and if only the characteristics and limitations of various data structuring formalisms are known.

In order to act intelligently the system should have 'knowledge' about knowledge:*meta-knowledge*. Also, we need a mechanism to retrieve information through variety of means depending upon what user finds convenient at that point of time. We also should provide mechanism to change levels and granularity of the unit under consideration. The organization of information should be transparent to the user.

If we want system to be flexible, retrieval should be possible in many ways: deliberate, through meta-knowledge , multiple scanning and searching.

One important factor that should be taken into account, while describing data descriptors is their semantic content, which can be used to guide queries.

1.3 Importance of an Interpretational Base - CIBA

In *Modelling Language Behaviour* Narasimhan has expressed the need for an interpretational base to make communication among community members possible. We quote it here:

We have seen that the pragmatics of language behaviour relate to a world of behaviour environments on the one hand, and to an agent considered as a behavioural system on the other. The interlinking of the structures that support language behaviour with the structures that underlie behaviour in the other modalities results in an interpretational base. And this interpretational base enables an agent to describe, manipulate, and explore in the language modality the aspects of the world around him, and also his own agentive aspects.

It is precisely this interpretational base that a child builds up through well-defined interactions with adults when he acquires his first language.

— Narasimhan[Narasimhan, 1981]

CIBA plays an important role in providing *the interpretational base* for knowledge-driven systems.

The overall organization of the thesis is as follows:

1. Introduction
2. Biological Information Systems
3. The World of Computers: Review
4. Design Considerations
5. Singlish - Streamlined English
6. Conceptual Information Base 'CIBA'
7. Text-based System for Conceptual Knowledge
8. Modelling Factual Information
9. ReWIre - Real World Information Retrieval
10. Implementation
11. Conclusion

Chapter 2

Biological Information Systems

2.1 A View of Biological Information Systems

Humans can predict, explain, design and control, all without benefit of science, much less theory. Perhaps what best characterizes science methodologically is its ability to get these activities into external symbolic artifacts, available to all who are "skilled in the art".

— Allen Newell [Newell, 1990]

In the previous chapter, we had a brief survey of computer systems. We have also seen that if queries are to be answered at what we have been calling the computational level, some principled way of representing knowledge must be found as the representation of knowledge has a bearing on what can be answered. The nearest and more or less perfect model for the knowledge representation that can be aimed at, is the human mind. However, the nature of information we carry in our head is not still very clear to us. We know a lot about various "things", about

their various "aspects" in various "ways". In the physical world, these things can roughly be classified as objects or actors or events or states. Various aspects can be classified as their characteristics or attributes which have various dimensions like time, space, colour, participants, etc. To arrive at the model of the biological information system, let us position a human being living in this world in the centre and start analysing things from his viewpoint.

The two *worlds* of importance to him are the internal world (*mental world*) and the *external world*.

He being a living creature, has a *goal*, a mission to survive not only as an individual but as a species.

He has basic needs: food, shelter, reproduction and happiness.

His behaviour can be described in two steps:

- Perceive(*input*) the surrounding
- Act(*output*) to achieve his goals

Thus he is a *performer* (an *actor* or an *agent*) in this world. What he goes through is his *experience*.

His performance improves by *experiencing*. He *stores* or *memorizes* his experiences using some *patterns*. In his lifetime, he comes across various situations - he perceives them, compares them with *stored patterns* and *performs* accordingly. He thus *learns* from *experience*.

Pattern-matching can be considered as a basic operation in his internal world.

The technique he follows for it is *categorization*. Categorizing things into say good and bad, or friend and foe helps him in *acting*.

Now, let us look at the external world which surrounds him. The world has a *static aspect* and a *dynamic aspect*. As far as *static aspect* is concerned, the fundamental distinction can be made between living things like: *creatures* and *plants*, and non-living *objects*. *Creatures* and *Plants* are of many kinds.

The non-living things in nature are landscapes, viz.: mountains, valleys, and planes, water reservoirs like: rivers, lakes and oceans, various substances like salts, metals, soils and objects like rocks, pebbles.

He perceives the *external world* through his sense organs. He also finds a “kind” of his own. There are *entities* around him which are not objects yet he can feel their existence (heat, light, force etc.).

He perceives the *dynamic* aspect of the world through *changes* in the world. He also differentiates between the changes initiated by *agents*(we call them *acts*) and changes not initiated by agents (we call them *happenings*).

Thus *Objects, Entities, Creatures(Agents), Plants, Acts and Happenings* can be considered as some of the *primitive categories* in *his world*.

2.1.1 A Need to Communicate

His instincts make him go in search of food and he starts forming various *categories* like good food and bad food. He survives on *plants* and *other creatures*. He also looks for shelter to protect him.

To make things possible for him, he uses *tools*.

He has some special feeling for his species, he also has a need for entertainment and reproduction. He needs *co-operation* from members of his own kind(people).

Thus he faces the problem of “How to inform?”

He starts communicating through gestures to start with. However, as pointed out by many, this communication is limited to just “here and now”. He also needs *co-operation* at mental level, for learning. He needs an ability to share experiences, feelings and thoughts. And he discovers a tool, viz.: *language* for this *socialization*.

2.1.2 Identifiers and Classes

If the thing he is talking about is present, he can point at it. In its absence, he has to *name* it, to *identify* it. There is a necessity for naming various objects. However, if each thing is given a name, there will be too many names, to be known mutually. He finds it convenient to categorize things based on similarity and family resemblance, and refer to them by their class names; they become *Primitives*. He uses the *naming* technique sparingly, for instances of *things*.

2.1.3 Attributes

This human can now talk about anything as belonging to the class. However, to make it identifiable or unique in a given context, some extra things are to be specified about that thing, we call them *properties*. Thus fat, warm, simple etc. are properties of various things. In communication, how to convey maximum information with minimum efforts is always a concern. So he divides the properties into orthogonal groups in such a way that properties in one group are more or less independent of properties in another group and properties within the group are mutually exclusive, we call these groups *attributes*. The attributes colour, size, shape, weight are examples of such attributes. The attribute *colour* contains members red, black, yellow etc. The advantage of dividing, properties into such groups is two fold:

- Specifying one of the properties in a group, eliminates object's chance of having any other property in that group.
- For the same object, one can specify many properties, each from a different group.

2.1.4 Relationships

Many times, he feels a need to talk about not individuals but the *structures* of which individuals are *elements*. Various kinds of meaningful

connections among *elements* are captured, we call them *relationships*.

Relationships are

- Spatial (corresponding to linguistic symbols: *on, after, away*)
- Temporal (corresponding to linguistic symbols: *since, after, when*)
- Taxonomical (corresponding to *is-a* in computer science)
- Partonomic (corresponding to *part-of* in computer science)
- Possessive (corresponding to *has-a* in computer science)

Thus *properties* and *relationships* are two more primitive categories

2.1.5 Grammar

Using all these symbols, he has to form a *description* and also make sure that other *person* gets its *meaning*. This, he realizes, will be possible only if he follows mutually known conventions, a system, we call this system *grammar*. The ability to *describe* things in turn gives rise to higher level *complex concepts*.

2.1.6 Learning to Communicate

He sees to it that the newer members of the *community(children)* are initiated into this system. Once these *community members* are *initiated* into this system, *progressively* they get *familiar* with concepts and grammars and they start using and understanding *descriptions*.

2.1.7 Domains

The *exchange* of information with fellow members of his class makes him *knowledgeable*. (Knowledge is the information he has that he can

put to use). As knowledge increases, he categorizes it into various domains like *Business, Religion, Sports, Travel etc.*

2.1.8 Artifacts

His progress leads to *creation* of another kind of objects, (*artifacts*) to serve some purpose. They include *tools, machines, locomotives and structures like buildings.*

At mental level also, there were categories for various feelings, opinions, states etc.(This becomes our *mental plane.*)

His language acts as an interface binding the two world, *internal* and *external.* The objects created using a language give rise to another world - the world of information(This becomes our *discourse plane.* The objects in this world are also categorized into classes like words, utterances, sentences. (One categorization of sentences according to purposes they have to serve is called *speech acts.*

2.2 Role of Language in providing Second Level Representation

Use of a language makes many other things possible. One of the two tools that was responsible for the revolutionary progress of mankind is *language.*

Language is a symbol system that works within human capabilities. Language provides us with the following:

- A set of “meaningful” symbols
- A set of ways of forming “meaningful” compositions using these symbols.

Language builds our conceptual frame of mind. The key to abstractions or categorizations or groupings made by humans is through language as *abstractions, categories and groupings of interest are retained in the language as lexical units(words)*.

The words of a language are stable, universal, meaningful and atemporal *concepts* for the community that uses them. They make the social interaction possible among people. We also know words are optimization devices. It is said "A picture is worth thousand words". Certain words have such powerful associations that we can say "Sometimes a word is worth thousand pictures".

Here by doing some reverse engineering, we try to identify capabilities of the brain from what it does. We would like to put forward some conjectures:

Conjecture 1

Ability to build a structure, name it, store it and get it as one single unit must be fundamental to the brain. Natural language development must have taken place accordingly.

Conjecture 2

Human brain retrieves things in two ways:

- **By direct, straightforward retrieval**
- **Through retrieval of a symbol or a set of symbols which opens up a whole new structure called *chunk* which in turn unlocks further information.**

Conjecture 3

The necessity of naming is essentially the outcome of this architecture.

(Note: The idea of *chunks* is basically derived from these three conjectures. Researchers have earlier used this word to describe human memory. For our purpose, we define *chunk* as any structure consisting of two parts: *title* and *contents*. *Title* serves as a *handle* to pick up *contents*. Chunks can be organized into structures recursively.)

Conjecture 4

Human memory is basically of four kinds: short term, long term, buffered memory and priority memory.

One division within human-memory-storage which has been extremely influential is a distinction between short-term memory (STM) and long-term memory (LTM). An example often given is the difference between remembering a telephone number you have just looked up for long enough to dial it and recalling an incident from your childhood. Miller [Miller, 1956] in a very influential paper *The magical number seven, plus or minus two* points out that it does not matter what size the chunks that have to be stored in STM are. Apart from short term and long term memories, we assume two more kinds.

Priority memory is the memory which holds a person's interest, fears, ambition, goals etc.

An introduction is perhaps in place for the *buffer memory*. Human being in conscious state is continuously experiencing the world. He is observing, trying, analyzing, learning. He learns various skills like drawing, dancing, singing, playing. He learns by mistakes and by being told about various things. The experiences collected during the conscious state which typically last for a day, are put in the *buffer memory* as they come i.e. they are ordered temporally.

The human system alternately goes through the conscious state and the unconscious state (hybernation). In hybernation, the system has available, immediate experiences, i.e. what it has gone through, and it tries to assimilate these experiences by putting them in "proper place" with some triggers or index terms which eventually are used as "handles" to pick the information in future. At times new rules are formed,

new ideas are created, earlier ideas are strengthened, weakened or modified, and so on. ¹

In the conscious state, the store can be accessed only through triggers and indices. It is only in *unconscious* state that we have much more exposure.

Long term organization is achieved by picking things that are new and putting them in proper place.

Two supporting evidences for the *buffer memory* are:

- People write diaries at the end of the day, just before going to sleep, and they can capture the day's activities in chronological order in minute details. However after 4-5 days, it is almost impossible to remember those details.
- Last minute preparations for examinations help students a lot!

(Note: This idea should be incorporated in the knowledge-driven systems. The structures built should have expiry time depending upon their usefulness. We make it a feature of the information structure.)

We assume the human information storage to be basically semantic. The nature and structure of incoming information and the contents of priority memory affect the storage organization. There are various clusters corresponding to lists, sets, and trees in the store. Access is thus restricted by the nature of the storage.

Conjecture 5

The architecture of human information system is general and uniform. Just as LISP has list as its only structure available, the brain has chunks!

¹Perhaps dreams are the bye-products of this information churning. The dreams may not be totally random. They may have something to do with what is in our *priority memory* - our interests, fears, ambitions etc. and the experiences we have gone through.

2.3 Importance of the Concept Base

Concepts are the building blocks of thoughts. Most of the words in a language essentially provide 'names' for the *concepts*. Words in the vocabulary of a language are more or less expressions standing for individual concepts.

2.4 Importance of Discourses - Knowledge Representation with the Help of Language

The mappings of knowledge structures at mental level that are directly accessible to observations and analysis are the methods and techniques of information organization which the world of printing technology has given us. (We, have proposed a methodology in Chapter 7, which uses the text-books on a subject as a resource to be used in the development of knowledge about the subject.)

Conjecture 6

The various kinds of structures we, human beings have invented like *lists, tables, trees, networks, frames, etc* are just compression techniques and optimization techniques (syntactic devices). *Rules, procedures, scripts, prototypes and episodes* are, on the other hand, semantic devices; they serve different purposes.

(In Chapter 8, we show how by treating structures as transformations performed on descriptions, we can associate interpretations with them.)

Conjecture 7

The structure of the input affects structure inside a knowledge-

based system.

(We use this principle in Chapter 7, while getting knowledge structures from printed text.)

2.4.1 Knowledge Structures and Incremental Knowledge

Knowledge is incrementally added to a system. Here I would like to compare information structures in the system to trees in a forest. A forest grows as follows; to start with there are very few trees distributed over a large area. As years pass by, the trees grow, change their characteristics. If environment is favourable, new trees come into existence. Information system is like a forest.

About the placement of incremental information we put forward the following conjecture:

Conjecture 8

Proper placement of information is guided by

- **Principle of maximum relevance**
- **Principle of minimum conflict**
- **Set of beliefs**
- **Principle of rationality**

(We suggest using *topic* and *aspect* of the description to get an approximate match using semantic distances among concepts.)

Information Search

Over the years, we have tied ourselves up with an architecture where the operands and operators are required to be fetched into the central processing unit to execute an operation. At times, the operations involve scanning the entire structures. Does it mean that there is a lot of transfer of knowledge structures in the brain? We don't think so. It will be a very inefficient operation. Long term memory is quite stable and its structures are too huge to be transported along with all their links for processing. The best thing will be to inspect them in their place. The operations required to be performed on knowledge structures aren't many compared to vast number of structures present in a human information system. To give an analogy, we use a search-light all over to get clues and when we get the clues, we look at those structures minutely to see if they are relevant. (In chapter 8, we talk about the *Two phase matching protocol* which essentially explains this phenomenon.)

2.5 Characteristics of Organization of Knowledge by Biological Systems

Our analysis of biological systems ends in characterizing them as follows:

- Information is organized into chunks of some kinds.
- Chunks are of different sizes.
- One should be able to handle the smallest unit (one property of one object).
- Retrieval can be in chunks - whole object, whole event etc.
- Not all chunks are equally accessible.
- The way information is stored determines what is accessible for given input.

- Information may be available in single step or multiple steps.
- Only a few handles are available for chunks therefore scanning and checking is required.
- Retrieval can be deliberate, through meta-knowledge , multiple scanning and searching or direct through proximity.

2.6 Functionalities Expected from a Symbol System

In “Computation & Cognition: Toward a foundation of Cognitive Science” Pylyshyn[Pylyshyn, 1984] claims that cognition is essentially computation. He used the word *cognizer* for both computers as well as human beings.

We quote him here:

- *One of the main things cognizers have in common is, they act on the basis of ‘representations’.*

How is it possible for a physical system to act on the basis of ‘knowledge of’ objects and relations to which the system is not causally connected in the correct way?

They instantiate such representations physically as cognitive codes and that their behaviour is a causal consequence of operations carried out on these codes.

- *What it would be to use such models in a principled way to provide rigorous explanations as opposed to merely mimicking certain observed behaviours?*
- *In order that a computer program be viewed as a literal model of cognition, the program must correspond to the process people actually perform at a sufficiently fine and theoretically motivated level of resolution.*

- *To conclude that the model and the organism are carrying out the same process, we must impose independent constraints on what counts as the same process. This requires a principled notion of strong equivalence of processes.*
- *Choosing a set of basic operations is tantamount to choosing an appropriate level of comparison, one that defines strong equivalence.*
- *Explanations are relative to particular vocabularies.*
- *Computation is the only detailed hypothesis available for explaining how it is possible for a physical system to exhibit regularities that must be explained as 'rule following': or even as being governed by goals and beliefs. Cognitive science rests on the foundational assumption that there exists a natural set of generalizations that can be captured by using such cognitive terms."*

To summarize the discussion, the symbol system should have

- Ability to represent a symbol
- Compose a structure out of symbols
- Should pick up a structure by name
- Find an appropriate place for the symbol
- Establish relevant connections with other symbols
- Do some operations on a set of symbols to get result
- Ability to represent a result
- Establish relations among symbols
- Compare two structures of symbols
- Extract symbols from a structure
- Search/select structure satisfying a criteria

- Decompose symbols of a structure

Most of our computer languages do provide these functionalities. Thus we can say that the functional architecture of a computer is sufficient to make it a system resembling cognitive system.

In the next chapter, we will talk about entities in the computer plane and whether the computer formalisms are adequate to represent the knowledge of the real world.

Chapter 3

The World of Computers: Review

3.1 A Brief History of Progress

Computers are hierarchies of abstractions built upon electronic instantiations of Turing machines. This is a new and startling leap in a fairly long intellectual quest. Symbol systems and thought have co-evolved ever since human beings boosted its collective intelligence by creating and communicating more powerful mental representations. The convergence of mathematics and technology that made computers possible was the genesis of a new phase in intellectual co-evolution.

—Jean-Louis Gassee[Gassee, 1990]

In the study of any 'world', some time must be spent trying to understand the fundamental entities and the relationships between them. It is difficult to do justice to all the computer-related work done during last 40 years, published in hundreds of journals and conferences. It can be summarised as **Computer technology works towards changing**

the role of a computer as a tool to the role of the computer as a partner to human beings. This partnership is achieved by increasing the capacity (speed, storage), widening the application coverage, providing explanations (theory), reducing size of the machine, improving connectivity, making it easy to design, program and maintain the systems by providing higher level languages, tools, support, and last but not least bringing them closer to human-beings (user-friendly interfaces, multimedia-applications).

The field of computer technology is divided into various sub-fields(*domains*): databases, graphics, networking, programming languages, computer architecture, operating systems, theoretical computer science, and cognitive science.

In this study, we will be primarily focusing on computer languages, and various formalisms that are used to manipulate knowledge.

3.2 Computer Languages

The languages in the *computer plane* can be divided into following kinds, depending upon the function they serve.

- Algorithmic Languages
- Representation languages
- Data Manipulation Languages
- Query Languages

Algorithmic languages are used to write *programs* to get some job done. FORTRAN, COBOL, Pascal, Simula, Algol, Bliss, Programming Language C are some of the examples of these languages. Programming languages provide operations to manipulate representations. Programming languages distinguish two major categories: data and program. (Data may be embedded in a program module or it is stored separately on the secondary storage.)

One of the powerful tools that programming languages use (which is the fundamental idea behind algebra) is the mechanism of a variable: a token that symbolizes the value of an unknown quantity.

Various programming paradigms make programming easier by advocating division of tasks into subtasks, by advocating structured programming, by providing facility for handling data depending upon the kind(type), and by hiding some low-level details and managing them internally.

Programmer, typically with a particular job in hand, presumably has some idea of the method that will be followed. A *program* is just a set of instructions. The *scope* of the instructions is limited to the world of computers only. For example, the instructions access various *cells* in the computer to get their contents, perform some operations on these contents and put them back.

Since, it is the programmer, who using his *mental plane* and *discourse plane* understands the problem (hopefully) and maps it as a set of instructions, the *model* of its behaviour is in the mind of the programmer. The program cannot be used to explain its behaviour, to explain why the instructions are to be carried out in one particular way and not another and therefore it is very difficult to find whether the system is doing what it is intended to do.

The only explanation to why the program does a particular thing in a particular way is that the human behind the scene has modelled the situation in that way.

The only criteria for correctness of such systems is their input-output behaviour. But input/output behaviour for an open-ended system like real world with infinite classes and infinite objects is very difficult to certify even after exhaustive testing. Thus explaining the system, proving that the system is correct, and modifying the system all the three tasks become problematic.

3.2.1 Representation languages

Knowledge Representation languages attempt to provide descriptions of real-world situations. Rule-based languages, First-Order-Predicate-Calculus-based languages, languages for writing production systems etc, are examples of these languages. These languages presumably allow us to represent real-world situation, the objects in the real-world, their attributes and relationships among objects as well as descriptions. They support *inference mechanism* to get implicit knowledge from explicit knowledge.

3.2.2 Data Manipulation Languages

Data manipulation languages, are supposed to work on data in a database. Database technology emerged to manage large amount of shared information using large, shared data structures, stored on secondary storage, accessed concurrently by multiple users and possibly distributed over several machines and locations.

These languages are generally provided with DBMS software. A Database Management systems (DBMS) is the software that allows one or more persons to use and/or modify the data stored more or less permanently. This software is different from simple programs because it gives users the ability to store and use data definitions. A major purpose of a DBMS system is to provide users with an abstract view. The system hides certain details about how the data is stored and maintained. This is achieved by defining three levels at which database may be defined and viewed, viz., conceptual, logical and physical.

The data stored in databases is generally large and accessed simultaneously by many users. DBMS systems search information which is stored explicitly, there is no reasoning; Information is formatted, regular, explicit; and queries are stereotype. There have been three database supported structures, viz., Hierarchical, Network, Relational

3.2.3 Query Languages

Various design alternatives exist to build user interfaces. Following are the different kinds of user interfaces with their relative merits and demerits.

- **Menu Selection:** The user reads a list of items, selects the one most appropriate to his task, applies the syntax to indicate his selection, confirms the choice, initiates the action, and observes the effect. If the terminology and meaning of the items are understandable and distinct, then the user can accomplish his task with little learning or memorisation and few key strokes.
- **Form Fill-in:** When data entry is required, menu selection usually becomes cumbersome, and form fill-in (also called fill-in-the-blanks) is appropriate. Users see a display of related fields, move a cursor among the fields, and enter data where desired.
- **Command Languages:** For frequent users command languages provide a strong feeling of control and initiative. The users learn syntax and can often express complex possibilities rapidly, without having to read distracting prompts.
- **Direct Manipulation:** When a clever designer can create a visual representation of the world of action, the users' task can be greatly simplified by allowing direct manipulation of the objects of interest. Examples include display editors, LOTUS 1-2-3, air traffic control systems, and video games. By pointing at visual representations of objects and actions, users can rapidly carry out tasks and immediately observe the results. Keyboard entry of commands or menu choices is replaced by cursor motion devices to select from a visible set of objects and actions.

Before commenting on the mechanisms provided in the computer world to represent knowledge, let us mention some of the theoretical underpinnings.

3.3 Theoretical aspects

3.3.1 Computability

Alan Turing invented what is now known as Turing machine and formed a test of computability. He showed that the most complex of algorithms can be reduced to manipulations on strings of symbols by a Turing machine. The Turing thesis is that an abstract computational engine (The Turing Machine) can compute anything that is computable.

What is interesting about Turing's work from our point of view is that, to derive these results concerning the limits and universality of formalization, it was necessary to understand the notions of proof and deduction in a formal system in terms of manipulation of symbol tokens or marks on a piece of paper, where the manipulation is specified "mechanistically" in a manner entirely independent of the way the symbols might be interpreted.

3.3.2 Relational Model

Database management is all about mapping the *informal* real world into some *formal* machine representation. Codd's great contribution was that he developed a useful *formal* model - formal and hence mechanizable and hence usable as a basis for our computerized database systems. But a formal model is only useful to the extent that it has some reasonable mapping to the informal real world - i.e. to the extent that its formal constructs correspond in some reasonable way to relevant aspects of reality.

The relational model is a way of looking at data- that is, it is a prescription for a way of representing data (namely by means of tables), and a prescription for a way of manipulating that representation. More precisely the relational model is concerned with three aspects of data: data structuring.

data integrity and data manipulation.

— Date [Date, 1993]

Among the numerous advantages of “going relational”, there is one in particular that is the existence of a sound theoretical base.

... relational really is different. It is different because it is not ad hoc. Older systems were ad hoc; they may have provided solutions to certain important problems of their day, but they did not rest on any solid theoretical base. Relational systems, by contrast, do rest on such a base... which means that [they] are rock solid.

Thanks to this sound foundation, relational systems behave in well-defined ways; and (possibly without realizing the fact) users have a simple model of that behaviour in their mind, one that enables them to predict with confidence what the system will do in any given situation. There are (or should be) no surprises. This predictability means that user interfaces are easy to understand, document, teach, learn, use and remember.

— Date [Date, 1990]

3.3.3 A Uniform Technical Framework - LISP

John McCarthy [McCarthy, 1971] designed LISP as a technical framework, for the construction of systems that build up a knowledge base incrementally and that reason with respect to that knowledge base. LISP is to a large extent a functional language rather than a statement oriented one. LISP has its fundamental basis in recursive function theory. There is a provision to define functions recursively.

The only data structure that is used in LISP is list. LISP was the first high level programming language in which the internal representation of the program is defined to be exactly the same as that of the data; this makes it possible for a program X to create and execute a program Y or to operate upon itself.

3.3.4 Resolution Principle

Robinson's resolution principle led to new and general-purpose theorem provers. Resolution is a rule of inference - that is, it tells us how one proposition can follow from others. Using the resolution principle, we can prove theorems in a purely mechanical way from our axioms. We only have to decide which propositions to apply it to, and valid conclusions will be produced automatically.

Resolution is designed to work with formulae in clausal form. Given two clauses related in an appropriate way, it will generate a new clause that is a consequence of them. The basic idea is that if the same atomic formula appears both on the left hand side of one clause and the right hand side of another, then the clause obtained by fitting together the two clauses, missing out the duplicated formula, follows from them.

3.3.5 The Knowledge Level Approach

The term 'Knowledge level' was introduced by Newell [Newell, 1982] to describe a system/agent as if it possesses certain knowledge. Without making commitments about representation or implementation issues more precisely, a Knowledge level model, according to Newell, is a model of behaviour in terms of the Knowledge and goals the agent has and the actions the agent can perform. The agent is driven by the principle of rationality, that is, it selects actions that it expects will lead to the satisfaction of its goals. Such a knowledge level model is essentially aimed at explaining why the agent behaves in certain ways.

Tasks describe what should be done (the goals, subgoal), problem solv-

ing methods describe how goals can be achieved and domain models describe knowledge that is required.

3.4 Knowledge Representation and Computer World

Knowledge and Representation are equally important concepts for designing and analysing knowledge-based systems. Knowledge describes what systems do, and representations are how they do it. Knowledge-based systems can be designed that utilize multiple knowledge sources partitioned into different types or levels of abstraction and multiple representations (specialized for particular inferences).

Knowledge is a description of the world (determines a system's competence). Representation is a way Knowledge is encoded(determines performance, speed and efficiency).

Knowledge can usefully be partitioned into multiple sources in several ways, including by level of abstraction and by type. Different representations make different types of inferences apparent.

Knowledge is a description of the what, how and why of the world.

Different levels of knowledge use different ontologies for describing the world. With multiple levels of abstraction, problems can be solved at the appropriate level of detail.

3.5 Formalisms for Representing Knowledge

3.5.1 Production System

Planner, production systems and other such languages, allow general assertions to be expressed as content-specific rules of inference. Pro-

duction systems have been proposed as general models of cognitive architecture [Anderson, 1983], [Newell, 1990], and as a way to represent human expertise in computer programs.

In production systems, knowledge is encoded in a large body of conditional rules (productions) that are invoked by the specific contents of working memory.

A production system consists of a set of productions, each consisting of a set of conditions and a set of actions. At each moment, the conditions of all productions are matched against the elements of a temporary working memory, and those productions that are satisfied, execute putting new elements into working memory.

3.5.2 Schema-based Formalisms

One way of dealing with data is to categorize and sub-categorize it. Types are summary descriptions that may be viewed as encoding the agent's belief that there are objects in the physical world that conform to those descriptions and that these descriptions may be used to make inferences about objects.

Thus types serve two important purposes. They help structure and organize the knowledge about tokens so that the "quantum" of knowledge remains within manageable bounds and more importantly, they provide the basis for inductive learning and the encoding of abstractions.

According to the psychological notion of a schema from which the frame construct has evolved, frame describes the semantic context of the concept it stands for. For this purpose, some of the slots which make up a frame, as well as their slot entries, refer to other frames.

These frames represent concepts the first frame is strongly associated with and this forms its semantic context.

Slots which do not refer to other frames simply stand for some property domains and their entries for actual properties.

3.5.3 Semantic Networks

Semantic networks express knowledge in terms of concepts, their properties and the hierarchical sub/superclass relationship among concepts. Each concept is represented by a node and the hierarchical relationship between concepts is depicted by connecting appropriate concept nodes via *is-a* or *instance-of* links. Nodes at the lowest levels in the *is-a* hierarchy denote individuals (tokens) while nodes at higher level denote classes or categories of individuals (types). Concepts get more abstract as one moves up the *is-a* hierarchy. Properties are also represented by nodes and the fact that a property applies to a concept is represented by connecting the property node and the concept node via an appropriately labelled link. Typically, a property is attached at the highest concept in the conceptual hierarchy to which the property applies and if a property is attached to a node C, it is assumed that it applies to all nodes that are descendents of C.

Inheritance is the form of reasoning that leads an agent to infer properties of a concept based on the properties of its ancestors.

Recognition seeks a concept that has some specified property value.

3.5.4 FOPC (First Order Predicate Calculus)based Systems

Logic was originally devised as a way of representing the form of arguments, so that it would be possible to check in a formal way whether or not they were valid. Thus we can use logic to express prepositions (Prepositions are statements that are either true or false), the relations between prepositions and how one can validly infer some propositions from others. Predicate Calculus is one form in which logic can be expressed.

Propositions about the world are expressed by describing the objects that are involved in them. In predicate calculus, we represent objects by terms. The relationships between objects are expressed using predicate

symbols. The program consists of collection of facts and rules. Each fact or rule states an independent truth, independent of what other facts and rules there may be. Inference mechanism is provided to derive other facts from the given facts and rules.

3.5.5 The Object-oriented Environment

Object-oriented approach is a software design and development approach incorporating several sophisticated and efficient mechanisms that provide an organizational framework for the development of large and complex software projects. In comparison to traditional structured programming techniques, the object-oriented programming improves development of software systems by facilitating better factoring of functionality and related data.

The object-oriented programming approach provides a level of abstraction for software design and development, where data is encapsulated in objects and messages are used to manipulate the data.

3.5.6 Connectionist Formalism

Connectionist networks are made up of active elements that are capable of performing simple processing. These units have very high 'fan-ins' and 'fan-outs' and communicate with the rest of the network by transmitting a simple value. A unit transmits the same value to all units to which it is connected. The output value is closely related to the unit's internal potential and is best described as a level of activation. A unit's potential is a function of the amount of activation the unit has been receiving from other units. All inputs are weighted and combined in a manner specified by the potential function in order to update unit's potential.

A network consists of a large number of units connected to a large number of other units via links. The units are computational entities.

3.6 Conclusion

The representation schemes we have described here cannot be used for representing knowledge as they are at altogether different levels of abstraction. We quote here Date who has written about the difference between the two kinds of modelling in the context of databases.

The term data model is used in the literature to denote two quite distinct concepts, at two quite different levels of abstraction (and it is this fact that accounts for much of the confusion that surrounds the issue.)

Type 1 data model is a formal system, involving three components, namely a structural component, an integrity component, and a manipulative component. These components can be applied to the problems of any specific enterprise or organization, note carefully, however, that a type 1 data model, in and of itself has nothing to do with any such specific enterprise or organization[Date, 1990].

The relational model is a type 1 model.

Type 2 data model by contrast, is a model of some specific enterprise or organization, i.e. , it is essentially, just a database design. In other words, a type 2 model takes the facilities provided by some type 1 model and applies them to some specific problem.

— Date C. J. [Date, 1993]

To build a knowledge-driven system what we need is a type 2 model. In the subsequent chapter, we will be presenting our views about existing representation formalisms both on computer plane and discourse plane.

Chapter 4

Design Considerations

4.1 Introduction

The distinction of toy versus real tasks is not solely the distinction between basic and applied research. Tasks taken from the real world, performed by intelligent humans as part of their working lives, carry a prima facie guarantee of demanding appropriate intelligent activity by systems that would perform them. It can be argued that such tasks are the appropriate ones for AI to work on, even if the goal is basic research.

—— Allen Newell[Newell, 1982]

We cannot continue to build programs that we hope will scale up. We must scale them up ourselves.

—— Schank R. C.[Schank, 1991]

The construction of large, knowledge-driven applications is a complex task that comprises a number of activities and involves various partici-

pants. Not only should the components -knowledge acquisition, knowledge representation and knowledge retrieval work well individually but they should work in co-operation with one another and also with the humans who are the creators, mentors and beneficiaries of these systems. To meet this requirement, it is mandatory that both the system and people understand each other's language. If they speak different languages, then either people have to understand the system's language or the system has to understand people's language. Currently, people are expected to understand system's language; not only system's language but also its implementation details.

Designers of knowledge-based systems should understand that people and tasks come first. The primary objective of a Knowledge-based system should be to supply relevant knowledge to users who ask for it, in a way they understand it.

4.2 What can we inherit from Biological Information Systems

We all know that symbol-processing systems of today are far from satisfactory. Their performance doesn't match their potential.

The failure of these systems can be attributed to a certain extent to the usage of ancient methods - the methods that were used to tackle the problems of a different kind. Earlier tasks handled by computers had a formal base. Underlying formalisms were known; they were mostly from well-disciplined domains. People explicitly knew the methods and therefore could write algorithms which could eventually be converted into programs.

Another cause of failure can be the insensitivity to the characteristics of both Biological Information systems and Computer-based systems. While describing the architecture of "the connection machine," W Daniel Hillis criticizes the computer systems by saying that they are ignoring the laws of physics. He emphasizes that they should take the principles of physics such as the "effect of localization" into consideration. We

do agree with him. However we feel we should not stop there. While mimicking the brain, in order to exhibit the flexibility with which it processes information, designers of computer systems should take into account the capabilities and functionalities of *living organisms* too. *Artificial life* is still beyond our reach. Thus we should take into consideration the tools developed by the living-organisms and the limitations of non-living machines like computers and see where and how we can improve the performance and make up for the "life" which they don't have!

Connectionist mechanism models brain at an all together different level, at physical level and not at the knowledge level. We cannot do the way things happen in nature, but we can understand (hopefully) the principle, follow a different path, and achieve the same results. The success of the formalism will depend upon how well it corresponds to the real-life situation.

4.3 Role of Language

Noam Chomsky[Chomsky, 1975] called language "The mirror of mind". Language characterizes the input and output behaviour of the human information system. Over the years, language developed as a communication medium, but according to us, **it is the role of language as a representation medium that has made the progress of mankind possible.**

Language builds our conceptual frame of mind. Words of a language are stable, universal, meaningful and atemporal units. They represent *concepts* for the community that uses them. They make the social interaction possible among people.

The conceptualizations provided by a language essentially provide a framework for the language-using community for information representation, acquisition and retrieval.

Our proposed framework - the *concept-base* essentially is expected to take this role for *knowledge-driven* systems.

We strongly feel, that natural language should be used not only at the interface level, but at the representation level as well.

4.4 Problems with a Natural Language as a Representation Language

However, using natural language at the representation level presents many problems. The problems can be characterized as problems due to ambiguity, problems due to fuzziness of symbols, problems with context sensitivity and problems with idiosyncrasies of the language (too many usages).

4.5 Alternative - Formal Languages

One way to avoid the problems due to natural languages was to use a different class of languages; *formal languages* with computers. These languages are not context sensitive, they are precise. People soon discovered different kinds of problems with them. These problems can be characterised as lack of expressibility for human beings, narrowing of domain, limited scope and lack of universality.

Formal languages do not adhere to the conceptual basis provided by natural languages. They have limited semantics.

4.6 Why Semantics?

The need for semantics for representation language is very much agreed upon. Without some concrete specification of the meaning of a notational convention, what is implied by an expression in that language is unclear, and the comparisons to other notational systems are impossible. Without this there is no independent way of knowing whether the

conclusions drawn by the program are correct or complete.

As Putnam says in *Representation and Reality*

A computer-based system represents a real-world situation. If the system is representational, so that regularities in its behaviour can be captured only by referring to the content of its representation, then the rules must have the property that those that apply to a particular code or state will appear to depend on what it is code for, or, to cast it in the terms I used "respect the semantic interpretations" [Putnam, 1988].

4.7 Natural Language as a Universal and Stable System for Representing Knowledge

Our work is based on the fundamental assumption that behind the apparent idiosyncrasies of natural languages, lies a fundamentally sound and stable system.

At the mental level, concepts are the building blocks of thoughts. Human beings verbalize their thoughts through natural languages. The basis of language, as a form of communication, is the mediation of agreed signs or symbols. Natural languages have words that name the concepts. Use of words for concepts makes it possible to give a social meaning to concepts. Communicating through language is an optimization technique. With the help of the stable units(words), a mechanism to compose structures dynamically using an agreed upon convention(grammar), and associating agreed upon meanings(roles) to the compositions, human beings are able to generate infinite descriptions using finite means.

To quote [Rosch, 1978]

Concepts are mental representation of classes and their most important salient function is to promote cognitive economy.

We feel words from natural languages, standing for concepts, should be the building blocks for computer-based systems as well. Composition techniques of natural languages must be studied and used in knowledge-driven systems.

After all, the natural language has been an effective communication medium for centuries for human beings. The presence of apparent ambiguities in the surface form of a language can be attributed to, among other things, *optimising transformations* that are performed by human beings when they use a natural language to communicate with other human beings. They optimise using contextual clues, using words in metaphoric way, using *short-cuts, etc*. **Natural languages were meant for communication among human beings and if we have to use them with computers, we have to formalize and streamline them.**

4.8 Why Formalization is Difficult

Formalization of a natural language is a difficult task, because we really do not know why we use a language in a particular fashion. We humans learn these languages only through usage, by living and actively participating in an environment where they are used. For computers it is not possible to learn the *conventions* and *connections* in the same way. If we want computers to be our *partners*, in the sense that they can have a dialogue with us, can store our information, perform some operations on it, and give us what we want when we need it, then *externalization* of the "convention" is mandatory.

We feel that a natural language becomes problematic for a computer system, because,

- Most of the context-sensitiveness a natural language gets is due to the fact that the most common words it uses have many meanings. Also words have multi-class memberships.
- Since a natural language is meant for human to human commu-

nication, whenever possible, it tries to take *short-cuts*.

- Situational context is taken into account and certain things are omitted.
- As pointed out by Narasimhan [Narasimhan, 1981], we have to take into account the *language in use*. Thus peculiarities of a language should be taken care of.
- The problem of getting semantic interpretation of a natural language utterance is difficult, because in a sentence, the structure is flattened.
- Sometimes identification of part-of-speech is difficult (especially in case of English) in absence of syntactic markers.

4.9 What Needs to be Done?

The best option to solve the problem thus hinges on meeting the following requirements:

- Formalization of Knowledge is necessary
- Earlier methodology based on ill-defined symbols (corresponding to variable names) must be replaced by something based on concepts. We see a need to provide a mechanism to convey a unique sense of a word from among several senses.
- Human and computer systems, their capabilities and limitations should be taken into consideration to arrive at the model.
- Externalization of knowledge is necessary. Generalization, abstraction and other connections should be provided. After all, concepts are not isolated units, they are inter-related. We should provide enough explicit knowledge about concepts to make them correspond to unique senses.

- Streamlining of a natural language is a must. We also see the need to add **structure preserving punctuations** and syntactic markers to a natural language to make it suitable for knowledge-driven systems.
- Existing resources must be made use of.
- Language usage must be studied to understand intensions of various constructs. We do believe that the primary function of language syntax is to help conveying the meaning of the sentence. Thus many of the so-called peculiarities, can be traced down to the efforts at disambiguating the meanings. (In chapter VI, we will support this with some examples.)
- We should build higher level concepts (terminological bases) in various disciplines on the top of this concept-base in modular fashion. Navigation through these bases will be made possible if we organize them properly.
- A knowledge representation framework should provide ways and means to the one who designs the knowledge base to state what it is all about in an unambiguous way and the same convention must be passed on to the users of the knowledge-base.
- Even when information is of factual kind, a language for lexical phrases to be used to represent the fact-base should be provided. The query language and the language for lexical phrases both should have the *same* interpretational base.
- It is not enough to know that *things* are connected, it is also necessary to know what kind of connection it is!
- If we want system to be flexible(human-like), a mechanism to retrieve information through a variety of means depending upon what users find convenient at that point of time should be made available.
- We also should provide mechanism to change levels and granularity of the unit under consideration. The organization of information should be transparent to the user.

- Knowledge is not only in symbols nor just in connections but in both. Therefore a need for “A Unified Model for Concept Structuring”.

4.10 Guiding Principles

The method we are proposing for the building of *concept-base* for knowledge-driven systems is based on the following hypotheses:

1. Development of language dictionaries and encyclopedae in existing forms is to facilitate the concept organization in the brain. Thus the structures that we find in dictionaries etc. explaining various concepts may be isomorphic with the structures in the brain. We observe that, dictionary definitions that capture *meanings* are given in various ways: by equivalent words, by class membership along with distinguishing features, by examples or by descriptions. All of them can be abstracted as a single relationship between the word and the text giving its meaning viz. *relevance*. Principle of relevance applies without exception and plays an important role in the organization of information.
2. Ability to build a structure, name it and store, and get it as one single unit must be fundamental to the brain. Natural language development has taken place accordingly. The whole terminological base can be built starting from the primitives and some basic structuring mechanism.
3. The basic structuring mechanism can be based on *chunks*! For our purpose, we define *chunk* as any structure consisting of two parts: *title* and *contents*. *Title* serves as a *handle* to pick up *contents*. The chunks can form a hierarchy.
4. The structures in incoming information influence structuring of the internal information. (For example we can spell the words only from left to right because that is the way we have acquired them.)

5. The structuring of internal information influences search for the information.
6. The incoming information is incrementally stored in the brain to make it maximally relevant to the existing structures and it follows
 - The principle of economy
 - The principle of minimum conflict
 - The principle of maximum relevance
 - The principle of tuning
7. There are two principal ways of retrieval: One is a low level process - through proximity - based on topology. The reflexive thinking or perception works more or less in this way. The other way is high level inferencing or deliberate retrieval which is based on knowledge about the knowledge - *meta-knowledge*.
8. Even the low-level simple queries which do not involve much problem-solving need a different mechanism to get the relevant information than syntactic pattern matching. Search based on syntactic pattern matching is of little help in answering queries.
9. Knowledge organization based on intensions is like well-organized text in text-books with titles and sub-titles and may help in getting relevant information.
10. While finding *relevant* internal structure(s) corresponding to the descriptions in query, two major considerations are of importance:
 - The organization of structure
 - Relationships among individual words or phrases that describe the elements in the structure

Chapter 5

Singlish - Streamlined English

5.1 Introduction

In a very real sense, language is just an extension of our culture, a happy convention for the purpose of mutual communication. This shared context is essential for communication, it enriches it, and it limits it. Language is not merely an outgrowth of our culture, a convention for living in that culture. It is a fundamental determiner of our perspective. Our thinking is linguistic. Language is not only a medium for external communication, but for internal communication, planning, decision making and problem solving at conscious level.

—— Hilary Putnam[Putnam, 1988]

In previous chapters, we have explained why it is necessary to adopt the conceptual system underlying a natural language. We have also seen that the compositing descriptions using stable, meaningful *concepts* is the only way to have an open ended scheme for forming descriptions. In this chapter, we will describe Streamlined English (Singlish), its need, its mechanisms and its advantages.

English is the world's most widely-used language. It is the only language which is spoken all over the world and hence can be considered as an international language. Therefore, we have selected English as a language to streamline. In this chapter,

In this chapter, we will discuss the problems we face while parsing when we use natural language as it is (language in use). To alleviate these problems, we suggest an approach of streamlining English to remove structural ambiguities. We will take typical examples from a collection of papers from *Semantic Interpretation and the Resolution of Ambiguity* [Hirst, 1987] and show that most of the problems get eliminated

Finally we will compare our work with the other methods and describe future work that needs to be done.

We rely heavily on 'A Comprehensive Grammar for English Usage' by Quirk [Quirk *et al*, 1985] for English grammar as well as examples. We are also using Modern English Usage [Swan, 1980] for typical usages.

5.2 Overview of Basic English

We all use grammar whenever we speak or write. Grammar is the system by which a language works.

English has several devices for putting words into meaningful combinations. The three most important ones are word order, function words, and inflections. Words fall into different *categories*.

The term *word category* has been normally understood to refer to the most general categories to which lexical items can be appropriately assigned.

Linguists have often made a descriptive distinction between function (closed class) words and content (open class) words. Function words do not have a strong semantic content, but mark the beginning of syntactic constituents or tie the constituents together. Content words do not usually fulfil these roles but contribute mainly to meaning.

Categories that fall into the closed class are *preposition, pronoun, determiner, conjunction, auxiliary-verb and primary-verb*.

Categories that fall into the open class are *noun, adjective, full verb, and adverb*.

Closed class items are also called 'function words', 'grammatical words' or 'structure words'. They stress their function in the grammatical sense, as structured markers.

Wh-words(*who, why, where, how, when, what, whose*) are used when it is not known before what the item refers to, and so it needs to be stated in full.

(for example: The place where Mary lives is London.)

5.2.1 Functions of Major Types of Words

The primary functions of the words belonging to various categories in a sentence are as follows.

<u>function</u>	<u>Type</u>
Naming	Nouns and pronouns
predicating(stating or asserting)	verbs
modifying	adjectives, adverbs
connecting	prepositions, conjunctions

5.2.2 Simple Sentences and Syntactic Roles

Using the words belonging to different classes, we can form simple sentences. Though sentences are divided into four major syntactic

types, viz. *declarative, interrogative, imperative and exclamative*, we will be dealing with only declarative sentences in this study. Sentences can be *simple* or *multiple*. Simple sentences are traditionally divided into two major parts: a subject and a predicate.

The subject is often described as the constituent defining the topic of the sentence - that which the sentence is 'about'.

Predicate is described as 'what is asserted about the subject'.

A simple sentence chiefly involves the elements having syntactic roles - subject, verb, object, complement and adverbial. These roles are called *Parts of speech* of a language.

Verb is the most central role in a sentence. It is easier to identify and it determines what other elements may or must occur in the clause.

Subject of the sentence is typically a noun (or np or a nominal clause) It occurs before the verb. It is obligatory (except in imperatives where it is implied). It determines *number & person* of the verb. *Subject* can be considered as the *topic* of the sentence. It typically refers to information that is regarded by the *speaker* as given.

Objects can be *direct* and *indirect*. *Objects* are normally nouns (or a nominal clause). In sentences, *Objects* follow *subject* and *verb*. If both *objects* are present, *indirect object* comes before the *direct object*.

Adverbs refer to the circumstances of the situation. They come either in the beginning or at the end, or just before the verb.

Complementation is a function of the word (or clause) which follows *subject, verb and objects*(if any), and completes the specification of a meaning relationship

Thus the five *roles*(functional categories) of clause constituents are

- subject (S)
- verb (V)
- object (O) - direct object (Od) and indirect object (Oi)

- complement (C) - subject complement (Cs) and object complement (Co)
- adverbial (A) - subject related (As) and object related (Ao)

By eliminating the optional adverbs which form the *background* of the sentence, seven major clause types are established based on the permissible combinations of the seven functional categories.

Major clause types are

- SV - intransitive
- SVO - monotransitive
- SVC - copular
- SVA - copular
- SVOO - ditransitive
- SVOC - complex transitive
- SVOA - complex transitive

(Note: The terminology is taken from [Quirk, 1985].)

The clause types are determined by the verb class(*subcategory*) to which the word belongs. Different verb classes require either different complementation (Od, Oi, Cs, Co, A) to complete the meaning of the verb or no complementation.

5.2.3 Mappings from One Category to Another: Verbals

We have seen what a basic sentence means. Often, modifiers, connecting words and verbals are used to embellish the basic sentence pattern. These words and word groups enable us to expand basic sentences with

details and to combine basic sentences in ways that show the relationships among ideas. Our writing gains variety, complexity, and -at the same time- clarity and efficiency when we use these words and word groups(*phrases*)[Quirk *et al*, 1985]. (A phrase is a group of words that is used as a single part of speech like subject, object etc. in a sentence.)

Before studying various phrases, let us look at verbals which are special verb forms that have some of the characteristics and abilities of verbs but cannot function as predicates by themselves. Verbs make an assertion. Verbals do not; they function as nouns or modifiers. There are three kinds of verbals: Infinitives, participles, and gerunds.

5.2.4 Phrases

Instead of having single words belonging to those particular categories (noun, verb etc), a **sentence** has noun phrases, verb phrases, adjective phrases, adverb phrases or preposition phrases. Each phrase is named after a class of words **which** has a primary and obligatory function in it.

The verb phrase and the noun phrase are considered the most important categories. The verb phrase symbolized as a V element, is the most 'central' and 'indispensable' part of the clause.

The noun phrase is important more because of its multiplicity of functions. It can function as any of the clause constituents except V.

Functions of Words in the Phrases in General

A phrase has a *head*, a central constituent, to which other elements can be optionally added. Apart from the head, three other terms designating broad functions of elements within a phrase are, determination, modification and complementation.

Complementation is a function of a part of a phrase (or clause) which follows a word, and completes the specification of a meaning relation-

ship which that word implies.

Verb Phrase - VP or V

Verb phrases compose of two kinds of elements - auxiliary and main verb. Main verb either stands alone or is preceded by upto four verbs in an auxiliary function.

The identification of the verb element in general presents no problem, as this element can be realized only by a verb phrase.

Verb phrases can be either finite or infinite. In a finite verb phrase, only the first word is finite, subsequent words in the verb phrase are nonfinite. In a non-finite verb phrase, all verbs are nonfinite.

Following are the verb forms in English.

<u>form</u>	<u>type</u>
base	finite/infinite
-s form	finite (is/are/am/were- finite)
-ing	nonfinite
past tense	finite
past participle	nonfinite

Figure 5.1 Verb Forms

5.2.5 Semantic Roles of Various Syntactic Roles

Every sentence describes a situation in which a number of participants are involved.

Semantic Roles Taken by Subjects

- Subject as an actor
For example: *He threw a ball from the balcony.*
- Subject as external causer, instrument, or inanimate causer
For example: *A stone broke the glass.*
- Subject as affected
For example: *He felt nervous.*
- Subject as a recipient (with verbs have, own, possess, benefit see, hear, taste, smell)
For example: *I have a dog at home.*
- Positioner subject (with verbs sit, stand, lie, live, stay, remain, stance words) For example: *I am standing here for a long time.*
- Transitive subject (with verbs carry, hold, keep, wear)
For example: *These pipes carry water to the city.*
- Locative/temporal/eventive subject
For example:
Yesterday was a holiday.
This jar contains coffee.
- Prop it subject. The word 'it' is used as a dummy subject, when we are talking only about circumstances or situation only like time, atmosphere, condition, distance.
For example:
It is 10 O'clock.
It is too windy.
it is not very far.

Semantic Roles of Adverbials

Adverbials refer to the circumstances of the situation (adjuncts, sub-juncts), comment on the form or content of the clause (disjunct) or

provide a link between clauses(conjunct). Adverbial can be an adverb phrase, a prepositional phrase or an adverbial phrase.

5.2.6 Co-ordination

Both co-ordination and subordination involve the linking of units. In co-ordination, the units are constituents at the same level of constituent structure, whereas in subordination, they form a hierarchy, the subordinate units being a constituent of the superordinate units.

A major difference between co-ordination and sub-ordination of clauses is that the information in a sub-ordinate clause is often placed in the background with respect to the superordinate clause.

Co-ordination can link simple sentences or it can link sub-ordinate clauses.

For example: *He asked to be transferred, because [he was unhappy], [he saw no prospect of promotion] and [conditions were far better at the other office].*

Co-ordinators can also be used to link elements which are parts of clauses, rather than whole clauses.

The minimum unit which can normally be co-ordinated is the word. *The general principle governing the co-ordination of phrase and words is that the cojoins must belong to the same category, formally, functionally and semantically[Quirk et al, 1985].*

5.2.7 Complex Sentence

A sentence is made up of clauses. Clauses are simple sentences connected to show some relationship among them.

The multiple sentence, consists of more than one clause. A compound sentence is made up of two or more co-ordinated main clauses.

In a complex sentence there is only one main clause and one or more

subordinations.

Subordination is asymmetrical relation showing a hierarchy

Subordinate clause == clausal unit (adverbial)

A clause that is not subordinate is an independent clause.

Subordinate clause can appear

- as a clause element of the superordinate(adverbial clause)
- constituent of a phrase
relative clause post-modifying a noun phrase

A clause where verb phrase is finite (refer to section) is called a finite clause whereas a clause with infinite verb phrase is called an infinite clause.

Refer to table 5.1 for verb forms.

5.2.8 Relationship between Category and Functional Role

Verb as a role (we call it verb phrase VP) is the most central role in a sentence and it is the verb that determines what other elements may or must occur in the clause.

Verbs as a *category* of words can be divided into three major categories, according to their function within the verb phrase; we distinguish the open class of full verbs such as 'cut' from the closed classes of primary verbs (be, do, have) and of *modal auxiliary verbs* (will, might etc). Of these three classes, the full verbs can act only as main verbs, the modal auxiliaries can act only as auxiliary verbs, and the primary verbs can act either as main verbs or as auxiliary verbs.

Modal auxiliaries are so called because of their contribution of meanings in the area known as modality (including such concepts as volition, probability and obligation).

Primary Verbs - be, do, have

The three primary verbs in English are 'be', 'have' and 'do'.

'be' functions as both auxiliary and main verb.

- The verb be is a main verb with a copular function.
For example: Vidya is a happy girl.
- It works as an aspect auxiliary.
For example: He is learning English.
- It also works as a passive auxiliary.
For example: She was awarded a prize.

'Have' functions as both auxiliary and main verb.

- As an auxiliary for perfective aspect, have combines with a past participle.
For example: I have finished my work.
- As a main verb, it normally takes an object, and has various meanings such as possession, relationship, health.

'Do' also functions as both auxiliary and main verb.

Noun

Nouns can be used as subject, direct object or indirect object of a sentence. Apart from that, noun can be used as a subject complement. For example: that man is a fool.

Nouns can also function as heads of preposition phrase. Nouns of style or material can be used as both attributive and predicative. For example: the concrete floor ...

The floor is concrete.

Adverbs

Adverbs function as a head of an adverb phrase with or without modification.

Adverb may function in the clause itself as adverbial (in the foreground), i.e. as an element distinct from subject, verb, object and complement or they can be used for describing the situation (background).

(Note: In English, the words that act as premodifiers to adjectives and adverbs are also classified as adverbs, we feel this is unnecessary. They should be categorized as something different say *admodifiers*. Example: very happy, too fast, so well, stupid enough.)

Adverbs are traditionally divided into four classes: adjuncts, subjuncts, disjuncts and conjuncts.

Adjuncts and subjuncts are relatively integrated with the structure. Disjuncts are used for evaluation, authority, comment. Conjuncts are used for connecting clauses. They are used for introducing adverbial clauses, relative clauses or nominal clauses.

5.2.9 English and Syntactic Devices to Help in Understanding the Meaning

The primary function of a sentence is to convey a thought. Let us now see, how we get meaning out of an ordinary English sentence.

- A sentence has a topic (subject) and a predicate.
- The sentence can also be looked at as consisting of foreground and background.

Foreground consists of syntactic roles S, V, O, A and C. Background consists of optional adverbs, adverbial clauses, and relative clause.

- In English, all identifiers (proper names) start with a capital letter. This is a very good syntactic device to differentiate them from language words.

- In English, verb phrases are generally identifiable, because of their form and presence of auxiliaries.
- Noun phrases can be identified because, in most cases, they start with determiners.
- Preposition phrases can be determined because they start with prepositions.
- A comma is used as a separator for parts of speech as well as for separating clauses, when the need arises.
- The subject generally comes in the beginning, and separated by other constituents by a verb phrase.
- A question is identified by the end-mark as well as the starting word, which is usually a wh-word or auxiliary. In case of questions, when the subject cannot be the first word, another unique place for subject is identified, between auxiliary word(s) and main word. (That perhaps explains why an auxiliary verb is a must for a question.)
- A command or a request is identified by the presence of a verb in the beginning. Exclamations are identified by the end-mark as well as starting word which usually belongs to some close-class of words.
- Clauses start with special conjuncts (who, when, where, etc). These words also act as separators between two clauses along with comma as a separator between two clauses.
- Hyphen is used for linking words. It is especially useful when a premodifier consists of more than one word.
For example: closed-class words, bed-ridden person, empty-headed man.
- The topic of a sentence can be anything, even any word of a language. In that case to make it clear that one is talking about the word as an entity in the information word and not as a normal

word of the sentence, it is enclosed in single quotes.
For example: When 'for' is used after 'take', it means ...

- Most multi-word adverbs occur finally so that they don't get mixed up with the foreground.
- A word or phrase that is used in a novel or special way (in a metaphoric sense may be) is put in quotes.
For example: The information is "hardwired" into the program.
- There are many ways of describing things. To avoid ambiguity, sentences can be paraphrased differently. Information is 'compressed' into a sentence, by using finite clause, infinite clause, prepositional phrase, premodifiers or complements.

For example:

The man who brings oil rang the bell.

The man bringing oil everyday rang the bell.

The man for bringing oil rang the bell.

The oil man rang the bell.

5.3 Problems in Parsing English

In English, sentence is a basic a unit of discourse. Sentences are simple or complex. A simple sentence consists of a single independent clause. A multiple sentence contains one or more clauses as its immediate constituents. Multiple sentences are either compound or complex. In a compound sentence, the immediate constituents are two or more co-ordinate clauses. In a complex sentence, one or more of its elements such as direct object or adverbial are realized by a sub-ordinate clause.

¹In some grammars, nonfinite constructions are considered phrases rather than clauses. We follow [Quirk *et al*, 1985], and treat them as clauses because they can be analysed into clause elements.

There is no clear correspondence between words, their grammatical categories, their syntactic roles(place) in a phrase or in a sentence and their functions within a phrase or within a clause.

Words in general have many meanings. Some words can belong to different grammatical categories. We have already mentioned that a sentence can be considered as consisting of foreground description and background where foreground has roles (S,V,O,A or C). The syntactic roles taken by elements in the foreground can basically be found by the word order, since we know they follow the order : S, V, Oi, Od, (A or C) However, the positions of these roles are not absolute. Each of these roles, in turn, can be a multi-word phrase with one or more clause associated with it. In the absence of syntactic markers for the roles and separators and linkers for parts of speech, processing English mostly depends upon human beings' ability to make 'sense' out of the construction.

We have seen earlier that the verb phrase operates as a V element, as the most 'central' and 'indispensable' part of the clause. Verb phrase is also easier to identify. It is the verb phrase that determines what other elements may or must occur in the clause. However, a verb can belong, in its various senses, to a number of different subcategories and hence enter into a number of different clause types. Thus the number of objects, adverbials or complements in the foreground is also not fixed.

For example the word 'get' is used in 5 sub-categories.

- *He will get a surprise(SVO).*
- *He is getting angry(SVC).*
- *He got through the window(SVA).*
- *He got her a splendid present(SVOO).*
- *He got his shoes wet(SVOC).*
- *He got himself into trouble(SVOA).*

Thus it is not possible to get the word subcategory uniquely.

For example the following sentence can be treated as SVOO or SVOC.

I found her an interesting partner.

can mean either

- *I found her to be an interesting partner.*
or
- *I found an interesting partner for her.*

The general problems with natural languages are due to the following:

- **Multiple meanings:** Most of the context sensitiveness a natural language gets is due to the fact that the most common words it uses have many meanings.
- **Multiple categories:** Also words have multi-class memberships. For example *after, as, before, since* and *until* are prepositions as well as conjuncts.
- **Optimization:** Since natural language is basically meant for human to human communication, whenever possible, it tries to take *short-cuts*. Human beings rely on the receiver's ability to understand the right referent for a word.
- **Context:** Situational context is taken into account. and certain things are omitted.
- **Many usages.** As pointed out by Narasimhan [Narasimhan, 1981], we have to take into account the language in use. Thus peculiarities of language should be taken care of.
- **Flattening of the sentence structure:** The problem of getting semantic interpretation is difficult, because in a sentence, the structure is flattened. There is a mixing of boundaries. A participant in a sentence can be a head word, which has a part to play as one of the roles in a sentence, or it is a modifier to one of the head words. Most often, it is possible to get the roles of the participants correctly if the modifying symbols belong only to one grammatical

category(adjectives in case of nouns and adverbs in case of verbs) and therefore can be recognized syntactically. However, when a noun is modified by a noun, it may create some confusion.

Similarly when the symbols for nouns and adjectives are the same there may be some problem.

Also, when there are multiple modifiers, or when the noun phrase itself is a list and a descriptor applies to both, there may be problem.

When there is a nesting of descriptions, there will be the problem of deciding the scope of a modifier.

The specific problems associated with English, then, are

- Identification of *roles* is difficult, as syntactic markers are not present.
- English has too many usages
- The same function words serve many purposes. Quirk[Quirk *et al*, 1985] identifies eight uses of the word 'over'.
- Multiple category membership: The same symbol(word) belongs to many categories in English. Examples are surprise, present, can, will, make, like. Thus one has to find which category is intended in a particular description by looking at nearby words, features of the sentence(like its complexity, number of clauses, number of verbs).
- Compound nouns and noun phrase descriptions have no clear-cut boundaries. In Sanskrit, compound nouns are written without any space in between the words that are compounded.
- In some cases verb forms for past tense and past participle are the same as base forms.

5.4 Necessity for Streamlining English

In general, a natural language is a relatively efficient and accurate encoding of the information it conveys. What makes it difficult to accept as a semantic theory is "ambiguity". However, ambiguity is not a feature of a language; rather it is a side-effect. Whenever possible, language makes an effort to differentiate between different meanings.

We hypothesize that

The primary function of language syntax is to help in conveying the meaning of the sentence. Thus many of the so-called peculiarities, can be traced down to efforts at disambiguating the meanings.

Some of the peculiarities can be considered as high level(multiple word) patterns, which become stable units in the language just as words have become and they should be treated like words. For examples, phrasal verbs and phrases like in spite of"

A few peculiarities are due to historic reasons. We have no explanation for them, and we need not stick to them. Irregular forms for past tense and past participles of the verbs again can be looked upon as techniques, to keep the word syntactically close to its base form, by keeping its consonants more or less same, and by changing its vowels. By doing so the length of a word is kept the same by avoiding the use of suffix '-ed' or '-en'.

In order to get the 'head' of the phrase towards the beginning, so that people do not lose track of it, English recommends following usage.

It is true of adjective and adverb phrases, as of noun phrase, that one-word elements tend to precede the head, whereas multi-word elements tend to follow it. (exception: indeed and enough)[Quirk et al, 1985]

Many English verbs (phrasal verbs) consists of two parts:a 'base' verb (like bring, take, come) and another 'small word' (like up, down, off, away). The small word is either a preposition or an 'adverb particle'. In some cases, the meaning of a two-part verb is simply a combination

of the meaning of the two words. Examples are come in, run away, walk across, sit on. In some cases, the first word keeps its meaning, but the second has a special 'intensifying' sense. Examples are break up, tire out. In other cases, the new two-part verb has quite a different meaning from the two separate parts: give up means 'surrender'.

In case of transitive phrasal verbs, the particle is separable. The particle is expected to appear not immediately after the verb but after the objects. This we feel again is a disambiguation technique. If the particle (often it is a preposition) follows the verb, and followed by the object, the object will be misunderstood as a part of preposition phrase if it comes immediately after the verb.

Example of a separable particle: *They turned the light on.*

Thus, natural language has mechanisms to make a sentence unambiguous for *human beings*. Human beings tend to choose the meaning that makes *sense* by considering, along with the syntax, the overall pattern, meanings of participating words, their categories, context etc. However, if we have to use the language for humans as well as machines, we feel that many of the sentences which are unambiguous for human beings may appear ambiguous for computers.

We postulate here that

Streamlining and disciplining natural language can make it a good semantic language. The requirement of compositionality can be met if the syntax of a Natural language can be used for semantic compositions in the streamlined language.

Instead of devising an altogether new language, which people have to learn from scratch, we select an existing natural language to start with and streamline it to suit our purpose

5.5 Streamlining English

We *streamline* English by providing syntactic markers for grouping, separating, linking and highlighting various elements. We also allow

alternate verb forms for words that have irregular verb forms.

5.5.1 Punctuations and Markers

The punctuations and markers used for streamlining are as follows:

- Clauses are separated by backslash.
- Relative clauses are enclosed between a pair of double-backslashes.
- A noun phrase always has a determiner. (We provide semantically empty determiner '@' whose only function is to separate a noun phrase.)
- In constructs using 'infinity to', *to* can be connected to the verb by tilde.
- In case of ambiguity, the head of a noun phrase is marked by following it by an up-arrow.
- Connections between head and modifiers are explicitly shown by putting 'tilde' between them.
- When a whole clause takes part in a sentence as one of the parts-of-speech, it is enclosed in back quotes.
- A part of speech can be enclosed in square brackets.
- A verb can be marked by a 'star' in front of it.

5.5.2 Inflections

We provide alternate inflections to base-forms of verbs. Base-form is the form to which, in regular cases, inflectional suffixes are added to make inflected forms.

We allow the following forms:

any base noun + s \Rightarrow plural
any base verb + s \Rightarrow third person singular simple present
any full verb + ing \Rightarrow present participle
any full verb + ed \Rightarrow past tense
any full verb + en \Rightarrow past participle

any adjective + er \Rightarrow comparative adjective
any adjective + est \Rightarrow superlative adjective

5.5.3 Examples of Singlish Text

We will now give some sample sentences, which we have picked up from the book *Semantic interpretation and the resolution of ambiguity* [Hirst, 1987] which are used as examples of ambiguities by various authors.

In our convention, these sentences pose no problems.

Examples:

the paper~will was destroyed.
(Paper and will form a compound noun.)

the soup~pot~cover~handle is red.
(soup~pot~cover~handle is identified as one noun phrase.)

put the block~[in the box] on the table.
(The prepositional phrase 'in the box' is linked to the block)

which years do you have cost~figures for?
(Cost~figures is treated as a compound noun)

the old *man the boats.

(The word 'man' is highlighted
to show that it is a verb)

'that deer ate everything in my garden' surprised me.
(The whole clause enclosed in backquotes is taken
the role of subject in the sentence.)

that~deer ate everything in my garden last night.
(The word 'that' is linked to the 'deer' as a determiner)

I know 'that will be true'.
(The whole clause is taken as
the object of the sentence.)

I told the girl 'that I liked the story'.
(The whole clause is
taken as the object of the sentence.)

I told the girl \\\ whom I liked \\\ the story.
(Relative clause is identified.)

I told the girl the story \\\ that I liked \\
(Relative clause is identified.)

I know 'that boys are mean'.
(The whole clause in backquotes is
taken as the object of the sentence.)

I know 'that Tom will hit Mary'.
(The whole clause in backquotes is
taken as the object of the sentence.)

I know the boy - that you saw .
(Relative clause is identified.)

'what boys do' is not my business.
(The subject of the sentence

is a clause that is identified.)

I know 'that~boy is bad'.

(The whole clause in backquotes is
taken as the object of the sentence.)

visiting~relatives\^ can be a nuisance.

(A compound noun is identified
also the head word is identified with an uparrow

the falling~block needs painting.

(A compound noun is identified.)

I *will go to the show with you.

(Verb is highlighted.)

'what boys want' is fish.

(The whole clause in backquotes is
taken as the object of the sentence.)

I know that~boy should do it.

(the word 'that' is linked to the
word 'boy' as a determiner.)

i know 'that boys should do it'.

(The whole clause in backquotes is
taken as the object of the sentence.)

a block rests on smooth~horizontal~table.

(The compound noun is identified.)

the large~student~residence blocks my view.

(The compound noun is identified.)

Bill said 'that Mary left yesterday'.

(The whole clause in backquotes is
taken as the object of the sentence.)

5.6 Comparison with Other Methods

In computer science, the search for formalism to record descriptions precisely ended in first-order predicate calculus. However, first order language of predicate calculus, we feel, is too artificial for human beings to work comfortably with.

Frame-based languages which have slots resembling the intensions of sentences like 'who', 'why' etc. may be alright for user interfaces, question-answering or short discourses. But we cannot expect human beings to understand the whole text written in this form at the speed they understand natural language text.

Efforts at disambiguating natural language text have been on for the last three decades [Hirst, 1987][Grosz, 1986]. We feel that the problem is, to some extent, with natural languages. Every natural language has its plus and minus points, therefore ideas from the other languages should be used to streamline them. English, likewise, needs an acceptable canonical form. It has lived with many exceptions and there is nothing to be proud about them. Every child going through the trauma of remembering all the past and present forms is not really essential. Future generation need not carry on with the mistakes of the past. And if the earlier mistakes are accepted as preserved as a part of language, then the spelling mistakes etc. of people for whom it is a 'second' language should also be pardoned.

At least people should accept Singlish (or something similar) as 'Secondary English', the language in which the communities for whom it is a 'second language' (and computers) can interact.

5.7 Advantages and Future Work

We strongly believe that the language for knowledge representation should be based on the conceptual basis of natural languages to make it a standard language(*interlingua*). If idiosyncrasies of surface structures of a language are removed and then used for knowledge representation, many other interfaces are possible for the same knowledge reservoir. Knowledge representation will be simpler to understand. Browsing through the discourse written in the natural language like descriptions will be natural. Streamlining essentially adds a layer of extra markers. One can easily get rid of the extra markers while presenting the text to a human reader (if he wishes so).

We feel that people, not trained in its usage, will not have much difficulty in adopting Singlish.

As it is, learning English becomes a burden, as with every new word, one doesn't have to just know its meaning, but also its spelling and its pronunciation. Phonetic English can be a step in the right direction for computer systems of future. Phonetic English can be based on the roman alphabets, spellings and pronunciations having a one-to-one mapping as in the case for Indian languages. We have devised a scheme for writing Indian Languages using Roman scripts[Irani & Ram, 1992] which we find very convenient. The same scheme can be used to write phonetic English.

Chapter 6

Conceptual Information Base 'CIBA'

6.1 Introduction

Mysteries of energy, the secrets of life and the nature of communication are all tied into the notion of how complex codes are constructed from simple elements.

— Claude Shannon[Shannon, 1949]

In the previous chapter, we have seen the importance of a language and how the language provides a framework for composing descriptions. We have also seen the mechanism a language uses: it has a vocabulary and a grammar for forming descriptions. We have also seen how streamlining language is helpful if it is to be used with computers. In this chapter, we will worry about why we need to build a *Concept Base* and how to build it.

6.1.1 An Ideal

In 1945, Vannevar Bush - described a work-station he called *memex* intended to aid scholarly research and writing. Memex was to be a device in which an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with speed and flexibility. It is an enlarged, intimated supplement to his memory. The question to pursue is whether we can really make this dream come true, with the existing theoretical frameworks.

6.2 What is Meant by a Knowledge-based System

A large knowledge base contains (representation of) knowledge about an application domain and knowledge of how to perform a task relevant to the application domain. To deserve their name, knowledge bases have to be endowed with semantics - i.e. with an account of what their contents say about the application domains, as well as with appropriate inference mechanisms compatible with this account.

6.3 The Need for a Concept Base

Before describing what our Concept Base is, we will justify the need for it by critically observing other representation planes.

6.3.1 Mental Plane of Internal Language

The process of classification, the recognition of similarities, and the grouping of organisms and objects based thereon dates back to primitive man [Mayr, 1963]. More recently, Lenneberg has argued that categorization must be the basic cognitive process.

Thus categorization by a principle, or the formation of an (abstract) concept is apparently prior to and more primitive than the association of a sound pattern with a specific sensory experience. And the abstractness underlying meaning in general... may best be understood by considering concept-formation the primary cognitive process, and naming (as well as acquiring a name) the secondary cognitive process. Concepts... are not so much the "product" of man's cognition, but conceptualization is the "cognitive process" itself.

— Lenneberg [Lenneberg, 1967]

Humans are presumed to have mental models of the physical world in their head. They have *items*¹ that correspond to members of different primitive classes of concepts in the physical world. Along with that they have *items* for their own *vernacular concepts*. Vernacular concepts are medium-specific. The *vernacular concepts* at mental plane can also be categorized as acts (decide, think), states (sad, angry), entities (hope, idea), attributes (good, easy) etc.

Let us now describe the next plane: the discourse plane of spoken languages.

6.3.2 Discourse Plane of Spoken Languages

Categorization of sound patterns and of objects and events in the real world is basic to learning a spoken language. This thesis was developed by Brown [Brown, 1956] who termed first language learning 'a process of cognitive socialization' involving 'the coordination of speech categories with the categories of the nonlinguistic world [formed at the mental plane]'. This discourse plane has a vocabulary that corresponds to *items* at physical level, mental level and of its own vernacular level.

¹We are using the word *item* for the representation or symbol that stands for members of various primitive classes, as well as for identifiers at any of the four levels of representation.



Items

nouse
tree
dog
walk

----Physical World



Vernacular Concepts

Mapping

hope
aim
think

Objective world - tree, dog, walk

----Mental Plane



Vernacular Concepts

Mapping - words out of Phonemes

"discuss"
"argue"
"loudly"

Objective world - tree, dog, walk

Mental Plane - hope, aim, think

----Discourse Plane of Speech



Vernacular Concepts

Mapping - words using alphabets

Paragraph
Comma
Chapter
Write

Objective world - tree, dog, walk

Mental Plane - hope, aim, think

Discourse Plane of speech -

discuss, argue, loudly

----Discourse plane of writing



Vernacular Concepts

Mapping - using Concepts from CIBA

DBMS
FILE
PROGRAM
OS
COMPILER
SOFTWARE

Objective world - tree, dog, walk

Mental Plane - hope, aim, think

Discourse Plane of speech -

discuss, argue, loudly

Discourse Plane of Text

paragrapn, chapter, write.

----Computer Plane

Concept Mapping through Representation Schemes

examples are utterance, remark, speech, discussion, talking, listening). The utterances at this level, correspond to *thoughts* at mental level, while *words*(barring function words) correspond to *concepts* at the mental level.

As Lenneberg has expressed, *words tag cognitive processes and it is words that make these processes seem more static than they actually are.*

Studies carried on pre-linguistic children show that an infant engages in sorting and grouping (categorizing) objects in a consistent way before he or she has acquired a language[Johnson-Laird & Wason, 1977].

6.3.3 Discourse Plane of Written Languages

This plane has *items* that corresponds to *concepts* at previous three planes in addition to its *vernacular concepts*. Examples of vernacular concepts at this level are. 'paragraphs', 'reading', 'writing', etc.)

6.3.4 Computer Plane

One important characterization, we have attempted in this thesis is that of the representation schemes. Representation schemes, internal language of mental representations, spoken languages and written languages are built incrementally, each one having its own *plane*, having what we call a "concept base". Each succeeding plane, provides a mapping for *concepts* from the *earlier* plane; in addition to that it has its own "vernacular" concepts. It follows from this that if computer systems are to be used for knowledge representation, the computer plane should also be incrementally built.

To make the computer plane truly representational, the mappings of *concepts* from all the three planes should be provided in addition to its own *concepts*.

Thus if we want a formalism to represent 'knowledge' on computers, it

need not and should not start from scratch, inventing its own vocabulary, but instead (1) provide a mapping for the concepts that are there in the previous planes, (2) provide a mechanism to form descriptions using these concepts (3) give 'meaning', interpretations or definitions for the additional vocabulary (like records, programming, etc.) and (4) socialize the convention.

6.4 When is a Representation Truly Representational

Various people have expressed their views on what should be the desirable characteristics of a representation scheme [Davis et al. 1993], Ray Jackendoff [Ray, 1984]. One of the main characteristics, a representation must have is "semanticity".

Semantics is concerned with the relation between the representation and the world being modelled. The representation should not be limited only to the *concepts* in its own plane. We feel that is what programming languages do! They operate in the world of computers which is isolated with its own vocabulary. We can say they lack 'semantics'.

6.5 Issues in Representation of Conceptual Knowledge

Words in the language have a very important role to play in the cognitive development of human beings. They are symbolic objects. Just as a human being understands the objects around him similarly he understands these 'symbolic objects' when other people in the community use them. Repeated reference to symbolic objects and their participation in the events in daily life gives him many "instances" of these objects.

Just as many instances of an object type creates a general concept of that object, these instances create a "concept" corresponding to those

instances in the mind of a person.

Here we work on an axiom that there is a mechanism (innate) in the brain by which generalization of specifics takes place.

We also work on an axiom that there is a mechanism to store actual instances as well as generalized instances. This mechanism works on the principle of cognitive economy, optimizes space, storing time, accessing mechanism, and accessing time. (note: Through the same mechanism, a person gets the rules of the game, i.e. the grammar of the language.)

It is this knowledge of concepts and grammars which by forming the "interpretational base" for the person makes communication and understanding possible.

For human beings the way to general concepts is 1) through a raw form, through specific knowledge as human beings live and come across various "experiences" in the world and they have a mechanism to "generalize" and 2) through a compiled form: through written text, audio, video and other teaching material. (Here we are considering only the text-books.)

Computer systems therefore must be provided with this general knowledge as explicitly as possible

What today's knowledge-driven systems lack is this "general" knowledge in the sense described above.

Our Conceptual Information Base 'CIBA' is essentially designed to provide a computer system with enough "conceptual knowledge" to act as an "auxiliary sources of information".

It is instructive to think for a moment about how we might equip a machine with a concept system.

My basic hypothesis is that the development of languages and terminologies has something to do with the structuring of information in the brain. Capabilities and limitations of human mind get mirrored into them. There are many languages throughout the world which have mechanisms to express almost anything that can be expressed in any

other language. What they differ in is the terminology, phrases and atoms - which are influenced by the environment, culture etc.

Giving terminology is an optimization technique. Concepts get built using other concepts and get a label. When that term is used, the whole picture comes to the mind of the listener/reader. Terminologies differ across different languages,

6.6 Conceptualization - a Mechanism to Optimize Information Storage and Retrieval

Conceptualization can be looked upon as an optimization technique. While talking about concepts, Rosch[Rosch, 1978] has expressed the following:

Concepts are mental representations of classes and their most salient function is to promote cognitive economy. By partitioning the world into classes, we decrease the amount of information we must perceive, learn, remember, communicate, and reason about. Thus, if we had no concepts, we would have to refer to each individual entity by its own name; every different table for example, would be denoted by a different word.

Another important function of concepts is that they enable us to go beyond the information given(Bruner, Goodnow, & Austin, 1956). Concepts are our means of linking perceptual and nonperceptual information. Concepts, then, are recognition devices; they serve as entry points into our knowledge stores and provide us with expectations that we can use to guide our actions.

A third important function of concepts is that they can be combined to form complex concepts and thoughts(e.g. Osherson & Smith, 1981). Presumably our understanding of complex

concepts is based on our understanding of the constituent concepts.

6.7 Need to Build a Concept Base

Knowledge-driven systems should capture the symbol systems as well as composition system(grammar) of the communication medium *language*, in order to maintain expressibility.

Hence the need to form a standard and make it acceptable. Various *ontologies*(*Ontology* is an organization of basic concepts) should be published which appeal to people's intuition about their supremacies. The fittest will survive.

Human beings if asked to explicitly list all the basic words in their native language will not be able to do so. Therefore, we must methodically build a lexicon for a natural language.

6.8 The Problem

The task of building a realistic lexicon for a natural language is formidable. There is no well articulated theory of what it should contain. Also the number of words to be dealt with is enormous. It would certainly be impractical and probably unproductive for us to set about constructing a lexicon by hand. Machine readable versions of published dictionaries(MRDs) is a potential source of lexical information for use by automated natural language processing systems.

6.8.1 Problems with Machine Readable Dictionaries

We have studied applications based on MRDs[Irani, 1990]. The problem with these MRDs is that they are produced with a human reader in mind and therefore make many assumptions from the point of view of

processing by machine; for example, an assumption that the user can understand definitions of words written in English. They rely heavily on the user's background and commonsense knowledge to retrieve and comprehend the information they contain.

Secondly this information is usually presented in an informal rather than systematic fashion and often rests on inappropriate linguistic models, from the perspectives of natural language processing.

However, if we use dictionaries like LDOCE[Procter, 1978] where the information is systematically and formally coded, the task will be much simpler. LDOCE is a full sized dictionary designed for learners of English as a second language and contains over 55000 entries. An entry is defined as a collection of one or more sense definitions. A sense definition is a set of definitions, examples and other text associated with one sense of a head. Sense definitions are presented in a language that is a restricted subset of English. LDOCE claims that all entries are defined using a *controlled vocabulary* of around 2200 words and that the entries have a simple and regular syntax. The entries also give grammatical properties of the words.

Example of an LDOCE entry is
shell (noun) :- a hard covering of an animal,
or of an egg, fruit, nut or seed.

The most important advantage of this dictionary is that it provides a starting point from which to work with senses or meanings. However attempts to extract the meaning of a word sense from its description in a dictionary and to convey this by means of an encoding in a formal knowledge semantics requires, before anything else, the formalization of this general knowledge (world knowledge), without which no useful interpretation of any particular definition can be achieved.

The problem with LDOCE sense definitions is that though LDOCE definitions use *Controlled vocabulary* of 2200 words, each of these words is many-way ambiguous. In LDOCE, sense definition is said to be given with respect to the *central meaning* of the word, but no indication is given as to which of the many meanings of the words in this core vocabulary are considered central. This dictates a need for writing

definitions that represent unique senses.

6.9 How to Capture the Unique Senses

We have already mentioned earlier that the key to abstractions or categorizations or groupings made by humans is through language as *abstractions, categories and groupings of interest are retained in the language as lexical units(words).*

The conceptualizations provided by a language essentially provide a framework for the language-using community for information representation, acquisition and retrieval.

Our proposed framework - CIBA - essentially is expected to take this role for knowledge-driven systems

6.10 Nature of the Concept Base 'CIBA'

Information is basically the description of various things at various levels of details.

Description is a unit of information in CIBA as well as in the information systems based on CIBA. Primary function of a description is to convey information about the real-world. A description is a n-ary relation among various concepts or concept expressions. Most of the words in a language essentially provide 'names' for the concepts. In order to capture a unique sense represented by the word in a language, we restrict its sense by providing other dimensions to it. These dimensions are primitive, base, plane and domain and category. Section 6.11 explains these dimensions. We feel that these dimensions are enough to pin down the meaning of a concept.

To start with, CIBA consists of *Basic concepts, Primitive Concepts, Concept Expressions and Complex Concepts, Concept Operators, and Concept Mappings. Concept Operators and Concept Mappings are used to*

What is a Concept

A word in the language, in a particular context, in a particular domain, in a particular plane and having a primitive word associated with it denotes a concept.

$$\text{concept} = \{ \text{word, category, domain, plane, primitive, basic} \}$$

Where

Word is a symbol (word) from natural language that stands for this concept.

Category corresponds to the grammatical category of English (noun, adjective, verb, adverb etc.) corresponding to this concept.

Domain is the subject in which this concept is defined like mathematics, biology, ..

Plane is the *plane* for which it is defined, "physical", "mental", "discourse" or "computer".

Primitive is one of the *primitives* from the primitive list.

Basic is the *basic word* associated with this concept.

Identifiers

Apart from the concepts there are *identifiers* in the information base. These can be labels for various instances of actors or places etc.

$$\text{identifier} = \{ \text{symbol, concept} \}$$

Where *concept* is the name of a *concept* to which this *identifier* is connected.

Concept Expressions

Concept expressions are formed using operators which have different meaning for different types of concepts.

Examples of operators are:

+ addition for objects-composition

- removal of a part

== approximately equal

Order relationship

Combinations (roles each one plays) e.g. book on the table

Attribute + object (object has that attribute) e.g. A kind person

Figure 6 2 Concept Representation

form *concept expressions*.

A description also contains *Identifiers* and *Numerals*. *Identifiers* are the names for things in the real world. They correspond to proper names in natural languages.

In an information system, descriptions are organized into higher level structures using *chunks* to serve some purpose. The *intensions* of these description structures are captured by the *titles* of the chunk which again is a set of *descriptions*. The *chunks* can be hierarchically ordered, in which case the structure among corresponding *titles* becomes the *skeleton* of the entire structure.

We have already mentioned earlier that a description is an *n-ary relation* among concepts. The relationship a *concept* plays in this description is called *the role* of the concept in the description.

One of the most common types of analysis of sentence contexts is known as *case grammar*. It was proposed by the linguist Fillmore [Fillmore, 1968].² The basic idea is to analyse sentences into 'cases' attached to the verbs.

The main cases are listed below:

- Agent: animate being who initiates action
- Instrument: inanimate entity which is involved in the action
- Recipient: animate being who is affected by the action
- Object: inanimate entity which is affected by the action
- Locative: the location or direction of the action

According to case grammar, the lexicon would define each meaning of a verb in terms of the cases it can take.

²This idea is not completely new in an Indian environment. Paninian framework designed more than two thousand years ago for writing a grammar of Sanskrit is based on the idea of cases. Refer to [Bharati et al, 1995].

Human beings can build information systems on top of CIBA using *Singlish - Streamlined English*. We have seen in the previous chapter that Singlish is basically English enriched by extra markers, to delineate parts-of-speech and clauses, and to make its syntax more formal. CIBA environment also contains an *interpreter* that converts information given in Singlish into *descriptions* and a *generator* that converts *descriptions* into Singlish.

CIBA thus provides a concept base for interpretation of the information in a knowledge-driven system.

6.11 Methodology

In this section we will explain the various dimensions *basic, primitive, category, domain, and plane*.

6.11.1 Basic Concepts

Basic concepts are the concepts a normal adult is familiar with. Basic concepts are the concepts universally known and generally have surface words representing them in a natural language. Our cross-linguistic study has helped us identify many characteristics of natural languages. We have taken as basic concepts, concepts associated with words which have equivalent words in all 14 Indian languages and English.

For a list of basic concepts see appendix V.

6.11.2 Primitives

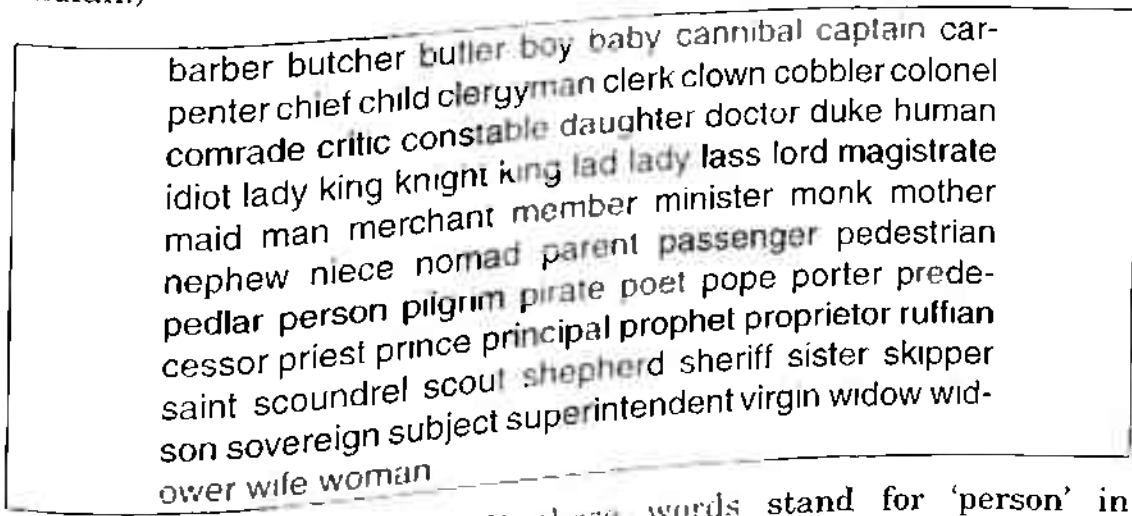
Primitives classify *concepts* into partitions. Aristotelian partitions that tend to be fundamental to all documented category sets and which we regard as partitioning the word at the broadest level are termed as *a priori primitives*. We use Aristotelian categories (with a few extra

categories) as a partitioning mechanism for *concepts*. These partitions are *Act, Person, Object, Entity, State, Happening, Theme, Information, Time, Space, Property, Quality*. The *primitives* we have chosen are hierarchically ordered under one of these partitions. (See Appendix V for the details).

The best source to get the primitive categories is dictionaries. Information already digested, categorized, and indexed can be suitably used to systematically construct a substantial common-sensical computational lexicon. A lexicon derived in such a way will save considerable effort and still produce a resource consisting of the true nature of interrelationships among concepts.

(Note: The difference between basic concepts and primitive concepts should be noted. Basic concepts are selected based on familiarity while primitives because of their root sense. Primitive words stand for the classes. Other concepts are specializations or combinations of these primitive concepts.

For example, *crow* is not a primitive but it can be a basic word for an Indian.)



barber butcher butler boy baby cannibal captain carpenter chief child clergyman clerk clown cobbler colonel comrade critic constable daughter doctor duke human idiot lady king knight king lad lady lass lord magistrate maid man merchant member minister monk mother nephew niece nomad parent passenger pedestrian pedlar person pilgrim pirate poet pope porter predecessor priest prince principal prophet proprietor ruffian saint scoundrel scout shepherd sheriff sister skipper son sovereign subject superintendent virgin widow widower wife woman

Figure 6.1: Figure 6.3 All these words stand for 'person' in PLATAEU [Irani, 1991]

6.11.3 Concept-Operators

The interpretations of words in a dictionary contain primitive concepts as well as basis concepts. Apart from them the dictionary entries contain function words (prepositions and conjunctions). We call them *concept-operators*.

We define their meanings like

a is on b \equiv *on*(a,b)

a is after b \equiv *after*(a,b)

Notice in English the order is :

concept-operated-on operation another-concept-operated-on

This may not be the case in the Indian languages.

The operators are generally overloaded, in the sense that the meaning changes depending upon the kind of operands.

For example:

a book on the table

a lecture on mathematics

Other operators are *and*, *or*, *not*, *sameAs*, *oneOf*, *someOf* *noneOf* and *Modifier*.

The important point is that every operator has an unambiguous interpretation in natural language.

These operators are used to form concept expressions in terms of *basic words*, *primitive words* and *mappings* and *operations*.

To convey information, people use these *concept expressions*.

A *description* (a sentence) is a *relation* that exists in the information system (discourse plane). It is a unit of information. (See Appendix XV for more information on *descriptions*.)

6.11.4 Mappings

In a language, there are certain transformations, like noun to verb, verb to noun, noun to modifier or adjective to adverb.

These transformations give *mappings* between concepts. These mappings are essential, as they carry the semantic meaning of the word.

From one perspective, a marriage is clearly a relationship between two people. (sample query: Who was Elizabeth Taylor married to in 1975? From another perspective, a marriage is equally clearly an entity in its own right (sample query: How many marriages have been performed in this Church since April?)" [Date, 1994]

Natural languages have solved this problem by having two words 'marry' and 'marriage' which 'map' onto each other.

There are also *relationships* among concepts that correspond to relationships like synonyms, antonyms, etc. in natural languages.

There are certain suffixes and prefixes that modify the meanings of the concepts.

Refer to Appendix VII for some examples.

6.12 Concepts in the Computer Plane

As stated earlier, once the mapping between concepts in the discourse plane (text plane) and concepts at computer plane is established, the only other thing to be done is to represent the *vernacular concepts* of the computer plane.

Let us first discuss, what different *primitive classes* exist in computer plane and their correspondence with classes in the discourse plane.

Attempts have been going on for years in the realm of philosophy and linguistics to precisely define natural language concepts. Here we build on that work a computational model of the world. These concepts need

explanation. (To make the difference between the *concepts* at computer level and other levels, we use Capital letters for concepts at computer level.)

- **OBJECT and ENTITY:** The world is viewed as composed of things of two kinds, concrete things called objects and conceptual things called entities. Correspondingly there will be **CONCEPTs** of types **OBJECT** and **ENTITY** in the computer world.
- **ATTRIBUTE:** In real world object and entities have properties. In computer world, there will be **ATTRIBUTEs**, features assigned to **OBJECT** or **ENTITY**. **OBJECTs** or **ENTITYs** are known to us through their **ATTRIBUTEs**. **ATTRIBUTEs** take different values for different **OBJECTs**.
- **DOMAIN:** The values for a particular **ATTRIBUTE** come from a particular set of values called a **DOMAIN** for that **ATTRIBUTE**.
- **COMPOSITE CONCEPT.** **CONCEPTs** can be **SIMPLE** or **COMPOSITE**.
COMPOSITE concepts are made out of more than one **CONCEPTs**.
- **HEREDITARY ATTRIBUTE:** The **ATTRIBUTEs** of **COMPOSITE CONCEPTs** may be related to the **ATTRIBUTEs** of the **CONCEPTs** in their composition. These are called **HEREDITARY** attributes.
- **INHERITANCE:** We also say that there exists **INHERITANCE** relationship between the **OBJECTs** if an **OBJECT** or **ENTITY** takes **HEREDITARY ATTRIBUTEs** from another **OBJECT**.
- **MULTIPLE INHERITANCE:** **MULTIPLE INHERITANCE** is present when an **OBJECT** or **ENTITY** takes **HEREDITARY ATTRIBUTEs** from more than one **OBJECT**.
- **STATE:** The set of all values of **ATTRIBUTEs** is the **STATE** of an **OBJECT** or an **ENTITY**.

- **CLASS:** **ATTRIBUTE**s can be used to form **CLASSE**s for **OBJECT**s and **ENTITY**s.
- **SIMILAR:** **OBJECT**s and **ENTITY**s that possess the same **ATTRIBUTE**s are **SIMILAR** in some sense.
- **EVENT:** A change of **STATE** is termed as **EVENT**. An **EVENT** can be described as an ordered pair $\langle s_1, s_2 \rangle$ where s_1 and s_2 are states before and after the event respectively. **EVENT** can correspond to an *act* or a *happening*.
- **TIME** and **SPACE:** **EVENT**s can have associated with them **TIME** and **SPACE**.
- **COMPOSITE EVENT:** Two **EVENT**s can be combined into a **COMPOSITE EVENT** if first ends in a **STATE** that is the beginning of the second **STATE**.
Thus the composite event for events $\langle s_1, s_2 \rangle$ and $\langle s_2, s_3 \rangle$ is $\langle s_1, s_3 \rangle$.
- **HISTORY:** The effect of **OBJECT**s on each other manifests through the history of the **OBJECT**, namely the **STATE**s they traverse in time.
- **ACT:** **OBJECT**s and **ENTITY**s can **ACT** upon each other.
- **RELATION:** There also exist **RELATION**s among **OBJECT**s and **ENTITY**s.
- An **OBJECT** or **ENTITY** can have a **NAME** in real world.
- **IDENTIFIER:** An **OBJECT** or **ENTITY** can have an **IDENTIFIER** in computer world. An **OBJECT** or **ENTITY** if it has an **IDENTIFIER** keeps its **IDENTIFIER** throughout its **HISTORY**.
- **PROCEDURE:** Procedures are computer programs.
- **OPERATOR:** An **OPERATOR** can be a basic operator or a **PROCEDURE**.

- **RULE:** RULE is a restriction on the possible values of the **ATTRIBUTES** of an **OBJECT** or **ENTITY**. **RULEs** can be formed using **OPERATORs**.
- **SUBTYPE:** **CONCEPTs** can be categorized into **SUBTYPEs**.
- **SYSTEM:** A **SYSTEM** corresponds to a universe of discourse.
- **THEME:** The purpose for which the system is built is **THEME** of the **SYSTEM**.

Full knowledge of an **OBJECT** or **ENTITY** requires information about how the **STATEs** of the **OBJECT** and **ENTITY** can change and which **EVENTs** or **ACTIONs** change them. The computational objects belong to one of these primitives.

SYSTEM
 OBJECT
 ENTITY
 NAME
 IDENTIFIER
 DOMAIN
 ATTRIBUTE
 RULE
 STATE
 EVENT
 RELATION
 OPERATOR
 TIME
 SPACE
 THEME

In the chapters 8, 9, and 10 we will illustrate how the concept base can be used for *knowledge-driven systems*.

Chapter 7

Text-based System for Conceptual Knowledge

7.1 Introduction

Knowledge in large quantities organized into usable chunks is an essential ingredient of a great deal of intellectual behaviour.

Janet Kolodner [Kolodner, 1986]

In Chapter 6, we have seen how we can represent *basic concepts* using *dimensions*, and also how to represent *simple concepts* as *concept expressions* using *basic concepts* and *primitives*. (Most of the words in a dictionary come under this category).

In this chapter, we will investigate how to get knowledge about *Complex concepts* in the system. A complex concept corresponds to *terminology* in a specific subject or an encyclopedic entry. A *complex concept* is defined using a *chunk*. Once the initial concept-base with its *interpreter* for descriptions is ready (first order concept-base), we can make use of it for knowledge acquisition in various domains like Mathematics, Physics, Chemistry, etc. and store that knowledge in a modular fashion.

Here we rely on existing sources to get knowledge as well as its organization.

It is widely accepted that the effective development of knowledge bases depends heavily on its knowledge elicitation. We view knowledge engineering as a process of information transformation in which knowledge is acquired and ultimately transformed to a formal representation. We feel that the prescriptive frameworks currently outlined for knowledge engineering give little assistance on how to tackle the typical knowledge acquisition and representation problem.

7.2 Our Strategy

Although rapid advances in hardware technology have made it possible to store and access text of any size, a critical bottleneck has occurred in the knowledge acquisition phase.

It is very difficult to get adequate knowledge in structured form from the experts which the existing systems demand. Though we agree that the nearest and more or less perfect model for the knowledge structure available that can be aimed at, is the human mind, human knowledge structures cannot be observed directly. Alternative knowledge structures that are directly accessible to observations and analysis are the methods and techniques of information organization which the world of printing technology has given us. Here we propose a methodology which uses the text-books on a subject as a resource to be used in the development of a knowledge-base for the subject. Our work is based on the following hypotheses:

- To answer any general question, (not anticipated by the designer) requires deep knowledge. Deep Knowledge can be used to analyse and explain phenomena and findings of that domain. Deep knowledge is not easy to be acquired directly form domain experts
- The degree to which a program can make decisions rationally is entirely dependent on the degree to which it has access to and is

able to take into account, the relevant information, most of all, the program must be able to recognize relevant information as relevant.

- No knowledge representation scheme can claim the expressive power and compactness of the printed (raw) text, that is used by human beings.
- The books are common carriers of human knowledge. While being written by an author, the contents are always deliberately arranged and located in order to be easily read and understood. So the presentations in the book usually appear in compact and natural chunks.
- Books also give a pictorial support to make understanding easy.
- Another characteristic of a book is the latent organization of knowledge.
- For a text-based system, selection of raw text is a crucial factor to its success.
- The efficacy of a knowledge engineering methodology is diminished if it is not supported by software tools.

The methodology we follow is as follows:

- Conversion of books into machine readable form.
- Getting the logical structure of the book (organization).
- Selection of the *best* book using the selection criteria given below.
- Formation of the *skeleton* for the knowledge-base on the subject.
- Transformation of *skeleton descriptors* picked from the *raw text* into Singlish descriptions.
- Filling the *skeleton* using the original text.

Let us see what we can do to achieve this.

7.2.1 Conversion of Books into Machine Readable Form

Though character recognition is a hard problem, recognition of printed characters is fairly common-place[Nagy, 1989]. Conventions that are used to signify certain things in the text can be captured and used by the scanner to get the category of the text. (Note: We at NCST, also have developed a system for identifying characters and recognizing characteristics of documents. Our work is based on the extraction of features using Mathematical morphology[Gupta & Irani, 1994].)

Constituents of Books

The information provided in a text book has the following constituents.

- **Title:** Title of a book gives an overall idea on what the book is about and often about the level of the subject.
- **Table of contents.** Table of contents gives the structure of the presentation of the subject.
- **Index terms:** Index terms give the terminology of the subject as well as the position where each term is used in the body of the book.
- **Appendix:** Appendix consists of compilation of some important information (precise, factual, related).
- **Preface** Preface gives the information about the user model assumed and the depth of the subject.
- **Body of the book:** Body of the book describes the subject matter, is linearly arranged and divided into chapters, sections, consisting of heads, subheads, paragraph names, italic words, illustrations, diagrams, pictures, etc. (It also has visual clues to differentiate logical parts of the text.)

7.2.2 Getting the Logical Structure of the Body of the Book

The structuring of printed(paper) text is usually guided by the following considerations.

- Constraints imposed by the physical characteristics of the printed page
- Aesthetic and convenience aspects in the mode of usage of the book
- More importantly, need to facilitate easy comprehension of the concepts presented in the book.

Printed document, therefore, can be viewed as an application of structuring operations on the textual/graphical components included in the document. The structuring approach is dictated largely by the nature of the medium(in this case paper). Moving from paper to electronic medium offers unprecedented potential for structuring of textual/graphical information. Intelligent scanners can be used to convert the existing paper text into computer readable text using, say, some sort of *print analysers*. While text formatters use the logical descriptions of the concepts like title, footnote, etc. to convert the text into aesthetically pleasing and functionally organized printed pages, a print analyzer can use the page layout and its knowledge of formatting conventions to separate out the organizational information about the text(see Figure 7.1). While given the logical descriptions for concepts(like title, footnote etc.), text formatters transfer the text into a form that looks like a typical printed page(or even better), a *print analyser* will look at the page layout and with its knowledge of formatting conventions, separate out the organizational information about the text(See Figure 1).

4 Scheduling Concepts

4.1 Review of Multiprogramming Concepts

multiprogramming
throughput

Figure 4.1 Two jobs, A and B for execution

Figure 4.2 Job execution without multiprogramming

Figure 4.3 Job execution with multiprogramming

4.2 Scheduling Concepts

4.2.1 Basic Components

jobs
user programs
processes

CPU-I/O Burst cycle

cycle
cpu burst
I/O burst

Figure 4.4 Execution is an alternating sequence of cpu and I/O bursts

Figure 4.5 Histogram of cpu burst times

Process State

state
new, active, waiting or halted

Figure 4.6 Process state diagram

ready
running

Figure 4.7 Refined process state diagram

Process control block

Process control block
Figure 4.8 Process control block

7-1 **Figure 7-1:** Information Organization

Though the subject matter varies greatly from one text-book on the subject to another, most of the books qualify to be the books on good presentation of concepts in a particular subject.

What remains invariant across the different text books is the key-ideas in the subject which correspond to the key-phrases called the terminology of the subject. The key-ideas or the concepts 'break' down the subject into identifiable units. The terminology serves as labels or identifiers for the concepts and can, in turn, be used in describing other concepts.

Thus, taking terminology as the basis for a particular subject, we will present the following evaluation criteria for the text from text-books on a subject.

1. Retroversion
2. Delineation
3. Circularity

Retroversion:

In a physical sense, text in text-books is linearly ordered. Ideally a concept definition should not occur after the concept usage, and if it does, then we say that *retroversion* has taken place. Let δ be some small distance(threshold) allowed between the position of usage of concept and that of its definition in the text-book. (Position can be measured in absolute terms in words, lines or pages.)

Let θ be defined as

= 1 if (definition-position < usage-position)

OR (usage-position - definition-position) < δ

= 0 otherwise

We can then define Retroversion index(δ) for a book as

$$\frac{\sum \text{of } \theta}{n}$$

(Over all n)

where n is the number of concepts in the book.

usage-position is the first usage of the text.

definition-position is the position where the concept was defined.

Retroversion index is a means of measuring orderliness of the concepts. Retroversion index nearer to one indicates that the book has properly ordered the concepts while nearer to zero index indicates that the concepts in the book need reorganization.

Delineation Index

The associations among concepts are not often made explicit in the raw text; the text moves from one concept to another without making explicit boundaries.

Also, depending upon the broad category of the concept, human beings understand the text that follows, which implicitly gives information about various aspects of the text, viz. definition, explanation, comparison, examples, problems, shortcoming, usage, description, features, etc.

However, for a computer based system, the text with clear-cut boundaries and explicit demarkation is preferred. Let us call all the titles that signify hierarchy *skeleton descriptors*. Thus titles, subtitles, headings, subheadings, paragraph headings etc. are skeleton descriptors.

Let γ be the set of all skeleton descriptions in the text.
 Let n be the number of index terms in the book.

Let θ be a function such that

$$\theta = 1 \text{ if a skeleton description contains an index-term}$$

$$= 0 \text{ otherwise}$$

$$\Sigma \theta$$

over γ

$$\text{Delineation index(DI)} = \frac{\text{---}}{n}$$

Delineation Index is a measure of arrangement and separation of the text into modules.

Circularity

A concept is defined or explained using other concepts which in turn use other concepts and so on. Thus the set of prerequisites for a concept is very large (it is like part explosion in the bill-of-materials problem). However the surface definition of the concept contains only those concepts which are the direct prerequisites of the concept.

We have already stated earlier that there are variations in the ordering of concepts. Thus to get the ordering of the concepts, we consider not one particular text-book but the collection of several text books on the subject.

Let us introduce another function

$$\theta(i, j, b) = 1 \text{ if surface definition of concept } i \text{ contains concept } j$$

in book b

= 0 otherwise

$$\Sigma \theta(i,j,b)$$

(over all b)

$$\text{Circularity Index}(i,j) = \frac{\text{number of books considered}}{\text{number of books considered}}$$

Circularity Index(i,j) is the index for j to be a direct prerequisite of i.

Circularity of definitions exist if surface definition of i contains j and surface definition of j contains i

Suppose a matrix of prerequisites P(i,j) has elements such that $p_{ij} = \text{index}(i,j)$.

We should be able to find some permutation of concepts γ such that the matrix of prerequisites corresponding to this sequence is a lower triangular matrix. This sequence will then give the partial ordering among concepts. A book having a good concept-sequence will have the concept definitions in the same order as this partial order.

Derivation of Taxonomic Relationships

Hierarchical organization assumes one prominent relationship in printed text of concepts. This seems artificial as everything cannot fit neatly into a fixed place as there can be many hierarchies for the same concept depending upon different criteria as also there can be other types of relationships existing among concepts.

Human beings get the taxonomic relationships from the content pages and from the sizes, fonts and numbering of the *skeleton descriptors*. *Skeleton descriptors* are at different levels. For example chapter heading is at level 1, section heading is at level 2, subsection heading is at level 3, paragraph heading is at level 4 etc. The highest level starts with one.

Let us now try to derive the information about the taxonomic relationships among concepts.

$\theta(i, j, b) = 1$ if i is a concept and j comes under it but
in the lower skeleton descriptor in book b .
 $= 0$ otherwise

Index for direct taxonomic relationship between i and $j =$

$$\sum \theta(i, j, b)$$

(over all books)

Number of books having both the concepts

This information can be used to get the hierarchies of concepts for the text.

7.2.5 Formation of Skeleton for the Knowledge Base

Once the book is selected meeting the criteria defined above, the logical structure of the book can be taken as the *skeleton* for the knowledge base. As there is good correspondence between the logical structure of the book and organization of the knowledge in the knowledge base, the mapping of text from the book into the skeleton is comparatively easy. The skeleton can then be filled using the original text.

7.2.6 Transformation of Skeleton Descriptors

Skeleton descriptions are basically titles conveying intensions of the text that follows. They are either sentences or noun phrases in English. In Chapter 10, we have described an interpreter which can convert sentences or phrases into a canonical form. We have also described how skeleton descriptors can be useful while processing a query. The original text from the book in natural language can go in *skeleton descriptors* as it is or it can be put in the *canonical form*.

7.3 Conclusion

Text-based systems seem to be a plausible cost-effective and easier alternative to fully structured systems as they retain the expressive power and compactness of the printed text. In this chapter, we have suggested criteria for selection of the *ideal book* to provide *raw text* as well as *skeleton* for the text on a particular subject. The criteria apply to the books in a given subject with the same scope (i.e. number of concepts). Further research is needed in this area if we have to integrate text from different text-books in a single system.

Chapter 8

Modelling Factual Information

8.1 Introduction

Concepts that have proved useful for ordinary things easily assume so great an authority over us that we forget their terrestrial origin and accept them as unalterable facts. They then become labelled as "conceptual necessities". The road of scientific progress is frequently blocked for long periods by such error. It is therefore not just an idle game to exercise our ability to analyse familiar concepts, and to demonstrate the condition on which their justification and usefulness depend.

- Einstein

The purpose of this chapter is to stress the need to design a language for lexical phrases such as slot-names, relation-names, procedure-names, etc. which are used in Knowledge Representation systems. This language will have its basis in the *concepts* of a natural language, in the sense that all definitions or *interpretations* of the phrases can be given in terms of *concepts*.

The knowledge representation framework should provide ways and means to the one who designs the knowledge base to state what it is all about in an unambiguous way and the same convention must be passed on to the users of the knowledge-base. This will be possible if the knowledge representation language i.e. the language for lexical phrases and the query language (embellished natural language like say Singlish) have the same interpretation base.

In the previous chapter we have seen how conceptual information can be organized. In this chapter, we will consider modelling factual information. The word 'fact' according to Webster dictionary means, "anything that exists in reality" or "a truth known by actual experience or by observation". Factual information basically captures descriptions about various items(things) in the internal and external worlds of human beings.

Various formalisms are used to represent facts in computers like record-based systems, database management systems, semantic networks, and frame-based systems. In our view, no particular formalism is the 'best' for all kinds of information. We feel that the nature of data, the nature of relationships among the elements, and the purpose for which the data is used (queries expected) should together determine which formalism to choose in a given context.

We also feel that there should be an interpretational base provided for the information in information systems.

Thus, as stated by W A Woods[Woods, 1977]

A principal objective is to make the system sufficiently flexible that the decision maker can get information presented in whatever manner he finds helps him understand the situation, and to make it sufficiently intelligent and fluent that he can do this without having to take his attention away from the problem he is trying to solve and devote it instead to the issue of how to get computer to do what he wants.

Systems that allow natural language access to a conventional computerized databases are becoming available as software

products of today's markets. Such systems generally have limitations in the range of English syntax that they will understand and severe limitations in their discourse understanding abilities. Moreover, they are dependent on the generally artificial conceptualizations of the domains that are built into their data bases. In the next few years, such systems can be expected to evolve somewhat more sophisticated discourse understanding abilities, but still fall short of intelligently understanding what the user wants and responding appropriately. Systems with the latter capabilities will emerge as further progress is made in knowledge representation, modelling belief systems, and common sense reasoning.

In the following sections we will try to find answers to the following questions: What are the criteria on which to base the selection of structuring formalism? Do the criteria for selection have anything to do with the semantics of information as distinguished from the economics of processing and storing the data? We then discuss existing approaches and provides some comparative analysis. We emphasize computational treatment of the data model. It is argued that knowledge-based systems based on supports like CIBA and PLATEAU - offer considerable promise to remedy the deficiencies of the earlier, more adhoc, natural language interfaces.

8.2 The Problem

Factual information consists of descriptions for entities whose attribute values are recorded through those descriptions. Generally this information is structured using various structuring methods for grouping, categorizing, and ordering.

Due to the complex, evolutionary nature of applications and to the growing need for precision and semantic integrity information system designers are faced with increasingly more complex requirements. They face such problems as where to start, how to proceed, what rep-

resentation to use and how to model the systems for correctness and completeness.

The representation of complex knowledge and associative access to it lie at the core of factual-information systems. In these systems we try to organize data so that they represent the real world situation as clearly and naturally as possible, and yet are amenable to representation by computers.

Existing frameworks viz. graphs or semantic nets, schemas or frames and predicate logic based formalisms make extensive use of lexical items like slot-names. However the meaning of a slot in the frame or a link-name or a relationship-name is completely unconstrained and ill-defined.

Let us illustrate this by taking an example :

Consider a schema definition for a class of students with slots as follows:

1. name: Avinash Narwekar
2. address: C/3, Saijyot, Gupte Road, Dombivali
3. date-of-birth: 27 July 1961
4. hometown : Bombay
5. computer-languages-known: FORTRAN, C++
6. final-degree : B.Tech.
7. grade-obtained-in-final-degree: First
8. year-of-passing-final-degree . 1982

That the slots 6,7 and 8 are related can be derived by most of the human beings but not by the representation system. Thus the system will not be able to answer the question "In which year did this student pass his B.tech. examination?". The problem arises because the slot names play the role of variable names.

To the computational system, the difference in meaning appears only in the way various routines happen to manipulate these slots, that is, it is encoded procedurally, and therefore outside the formal system of representation.

Thus the basic problem in all these formalisms is the lack of adequate means to convey the meaning of terminologies, phrases used and their relationships in the design of an information-base.

Take another example of slot-names of an individual

1. friends-from-school
2. friends-from-college

A derived attribute(slot) can be *friend* (which is the union of the two slots). Thus on the one hand, there is the informal system of slot-names (from which human beings more or less can capture meanings) and on the other hand, is a query based on a natural language.

Another problem is related to structuring of information. The designers of systems choose one formalism or another, depending upon the familiarity and comfort and some intuition. They also try to make all kinds of information 'fit' into that formalism. This creates problems. Artificiality of the resulting data structures makes them difficult to understand, maintain and enhance.

Designers of information systems make a number of assumptions about the properties of data. These assumptions are not very often part of the computational system. Many of them are not even recorded. They are not known to the users; neither are they made available to the people who maintain and modify these systems.

Let us now take a look at some of the information structuring formalisms. Information structuring can be record-based, frame-based, in first-order-predicate-calculus form or in the form of semantic networks.

8.3 The Scenario

In formalisms based on logic, a knowledge-base may be considered as a collection of logical formulas. Advantage of such a scheme is the availability of inference rules to establish connectivity. The scheme facilitates knowledge-base descriptions which are in the form of assertions. First-order predicate calculus is a formal system that has as object language a first-order language, a set of axiom schemes, and two inference rules: modus ponens and generalization. Inference rule of modus ponens states that from p and p implies q , one can conclude q .

Robinson resolution principle is a rule of inference that permits a new clause to be derived from two given clauses; further, the derived clause is satisfiable (i.e. has a model) if the two given clauses are satisfiable

e.g. From

C1: $\text{not } P(a,b,c) \cup Q(d,e)$ and

C2: $P(x,y,z) \cup R(x,y)$

one obtains the derived clause

C3: $Q(d,e) \cup R(a,b)$

In procedural representation scheme, knowledge-base is a collection of processes implemented in a programming language like LISP. They have pattern-directed procedural invocation. The rules allow direct control and hence fast search.

In network representation scheme, a network represents knowledge in terms of a collection of objects (nodes - standing for individuals) and binary associations (directed labelled edges). These schemes support classification, aggregation, generalization and partitioning. These schemes address the issue of information retrieval since the association between objects defines access paths for traversing a network.

Databases are generally used to support the query and update of large amounts of regularly formatted data. They answer typical queries and are concerned more with the efficiency of retrieval and consistency and

persistence of data.

System designers intuitively choose one of the formalisms for information systems. Usually it is the characteristics of information that guides the choice of formalism. Some of the factors influencing information structuring are identified in literature. Nevertheless a systematic treatment is essential.

Let us now look at the characteristics of the information and how it influences information structuring.

8.3.1 Factors Influencing Information Structuring

The factors that influence information structuring are as follows:

- **Homogeneity:**

Record-based systems assume homogeneity of data. A *record* is a fixed sequence of field values, conforming to a static description usually contained in a data dictionary and/or in programs. The description consists mainly of name, length and data type for each field. Each such description defines one *record type*. Record structure presumes a horizontal and vertical homogeneity in data

Horizontally each record of a given type contains the same fields and vertically a given field contains the same 'kind' of information in each record.

Record structure fits best when the entire class has the same kinds of attributes. The more the information deviates from the norm of homogeneity, the less appropriate is the record configuration.

[Kent, 1979]

- **Vividness**[Levesque, 1986]. In *vivid* systems

1. there will be one-to-one correspondence between a certain class of symbols in the knowledge-base and the objects of interest in the world.

2. for every simple relationship of interest in the world, there will be a type of connection among symbols in the fact base such that the relationship holds among a group of objects in the world if and only if the appropriate connection exists among the corresponding symbols in the fact base.

Vivid fact bases, in some sense, are analogues of the domain
Object-oriented systems are more vivid than rule-based systems.

- **Completeness**

When a fact based system is forced to depend on incomplete facts, its ability to make decisions or solve problems is seriously compromised. In some cases, the lack of knowledge can be circumvented by using general defaults, while in other situations, special heuristics are required. However, no matter how a system plans to deal with incompleteness, it must first be able to determine where this incompleteness lies. In other words, a system has to find out exactly where knowledge is lacking before it can decide what to do about it. This suggests that a fact-base must be capable of providing information not only about the application area, but about itself as well. Thus, the language used to interact with a fact-base must allow a user to define and inquire about what the fact base does and does not know.

The reason incomplete fact-bases are so important is that, in many applications, the fact-base undergoes continual evolution. At each stage, information can be acquired that is potentially very vague or indefinite in nature. More importantly, a problem solving system cannot simply wait for the fact-base to stabilize in some final and complete form since this may never happen.

Logic-based systems provide mechanism to state incomplete knowledge. In record-based systems it is difficult to store incomplete knowledge except by providing fields with null values

- **Clustering**

Information is usually clustered. For example all the attributes of a particular entity are clustered in one group. In relational databases normalization technique helps in grouping information

in a particular way to avoid certain *anomalies* and to render itself amenable to mathematical treatment

Grouping or clustering also has influence on variations on the patterns of connectivity.

- **Inferencing**

When general laws are available as part of definitions of the data or as integrity constraints, systems with inferencing mechanism are preferred. The first-order-predicate-calculus-based systems provide this facility

- **Principle of localization**

When the data is localised or clustered in groups, a designer can model parts of an application independently(localized). This organization of information is possible in object-oriented systems

- **Principle of inheritance**

Many natural classifications are such that the entities are classified and sub-classified to several levels. All the classes which are specializations of the classes at the higher level inherit the attribute values for the common attributes from that level. Using the principle of inheritance a compact structure for these concepts can be given wherein only 'local' information is indicated i.e. at each stage only attributes that are of relevance at that particular level are given values. Thus repetition is avoided.

- **Principle of relevance**

This principle can also be stated as *principle for transitivity of semantic closeness*. When the information about the exact nature of queries is not known and when information cannot be categorized, a measure of closeness based on 'distance' can be formed. Semantic networks make use of this principle.

For example if a is connected to b, and b, in turn, is connected to c, and a and c are not connected in any other way, then we can say that a is closer to b than c.

- **Principle of abstraction**

At times information is a general statement connecting not indi-

viduals but higher-order items. This principle says that one need not say each and every fact that is true but can make a general statement about a class. Rule-based systems are based on this principle.

8.4 The Problem with Existing Formalisms

In the previous section we have seen the prime considerations behind choosing major data structuring formalisms. Let us now see if these data structures are adequate to meet our objectives.

- **Lack of methodology**
In some formalisms there is no methodology to follow while designing information structures. For example in semantic networks, because the network is simply a byproduct of the structure of terms in the language (the network is not itself a language), not all network-derived subsumption inferences are valid unless the hierarchy completely reflects all of the relations implicit in the descriptions in question. In other words, the descriptions must be in proper places in the network before any conclusions can be drawn.
- **Rigidity:**
Many of existing formalisms are record-based where data should be fitted into a relatively small number of predefined formats and they support requests of a relatively straightforward class. Forcing data into such formats usually leaves many things unexpressible. The artificiality of the resulting data structures often makes the expression of many kinds of request either impossible or a difficult programming task. The interfaces to the systems described above are very much implementation-dependent.
- **Lack of connectivity with the user interface**
Efforts are on to build natural-language-like user interfaces on top of the systems based on existing formalisms to make them

accessible to a wide variety of users. One problem which has been encountered when frame-like structures are used and natural language interface built on top of them is that of the selection of the appropriate data structures for a given input description from a query. How does a system with a large number of concepts choose a correct one? Sometimes particular words in an input description point directly to a particular data element, thus trivializing this problem. However more often it is the case that no one word in a text points definitely to a unique data element. A frame can be selected only by considering words in combinations or considering similar words.

- Lack of universality for users

All computational systems are in a sense symbol manipulation systems. Symbol systems imply universality. However the symbols used as names of the slots in a frame, or links and nodes in a network, or relation names and attribute names in databases, are completely unconstrained and ill-defined. These are good concepts in their own right but they tend to be completely unrecognized in these systems. The symbol names play the same role as variable names in programming languages. The difference in meaning appears only in the way various routines happen to manipulate these symbols, that is, it is encoded procedurally, and therefore outside the formal system of representation.

Except for situations in which the knowledge base models an artificial microworld, it cannot be assumed that the knowledge base is a complete description of the world it is intended to model. This has important consequences for the operations defined over a knowledge base, (inference, access, matching), as well as for the design methodology of Knowledge bases.

- Lack of descriptive ability for designers

Computer programming differs from mathematics in its interpretation of expressions like

$$A = 234 + 23 * 19$$

In mathematics, '=' is symmetric. Both the left hand side and right hand side, can be expressions and all valid operations can

be performed on both sides. The meaning of the expression does not change when we interchange the sides.

In computer programming (e.g. FORTRAN, COBOL), however, it is an assignment expression. Left hand side is just a symbol or an identifier or a label. Knowledge-based systems are totally handicapped due to this. Some mechanism should be provided to say

data description = value-expression

where data values come from the expressions using abstract data types and *left hand side* is an expression from linguistic domain with proper semantics associated to it.

The *data description* should be such that the following operations should be possible on it.

- Store a value in the element having this data description.
- Fetch the value from the element having this data description.
- Compare data descriptions for set-subset relationship
- Compare data descriptions for hierarchical relationship
- Compare data descriptions for similarity and approximate matching available for them.

The important thing to realize is
Frames, scripts, semantic networks, first order predicate calculus-based formalisms, connectionist networks, etc. are useful in cases where the system developer has the full knowledge of the sort of questions that will be asked to the system; these systems break down when a new or unexpected query is made.

8.5 Our Approach

We work on the hypothesis that if we use a language for data descriptions, built on a concept base say CIBA, we can give mean-

ingful interpretations to descriptions expressed in various data structures, by viewing them as descriptions corresponding to natural language, compressed with some intention.

We view data structuring as a *Compression technique*.

In this section, we will first list the advantages of having an underlying concept base for data descriptions. We will then discuss the need for *structuring* as well as the need for *clustering*. We will also discuss the advantages of having a framework. Finally, we will explain why we view data structuring as a *compression technique*.

8.5.1 Advantages of a Concept Base Underlying Descriptions

We advocate data modeling based on explicit phrases (not just words) and having same concept base as natural languages to help bridge the gap between the user interface and data structures.

The advantages of having *descriptions* based on a Concept base like CIBA are

- Humans will be able to design the data model easily
- Humans will be able to express the model easily.
- User interface will become natural and universal.

8.5.2 Need for Structuring

Structure is an arrangement or organization or the way in which parts are formed into a whole. Structuring of elements is not arbitrary but with some purpose in mind.

Structuring has one or more advantages:

- Structuring speeds up access to what is wanted. For example, a binary search tree can search in $O(\log n)$ time
- Structuring compresses data and thus saves storage space. For example, Frames store the common data with a prototype record.
- By storing data in modular fashion it makes it possible to change data. For example, the information about a student can be a single record.
- For human beings the data becomes more expressive. Individual descriptions can be structured so that relations among them mirror relations in reality. For example, a hierarchy can be easily understood in a tree form.

A structure is called a *discourse structure* if it serves a particular purpose.

Some higher level discourse structures are of the following types:

- definition
- example
- story
- explanation
- problem
- question
- answer
- method
- procedure

8.5.3 Need for Clustering

In a typical factual information system we find that

- descriptions about an object or an event or a state are grouped together
- descriptions of instances of a class of objects, events, states etc are grouped together

The descriptions also can be event descriptions, object descriptions, relations descriptions, actions descriptions or scene descriptions.

The advantage of this is that it categorizes information in bundles that can be looked at

8.5.4 Need for a Framework

Depending upon the nature of data and the kind of queries the information-base is expected to answer, the organization can be a database, a network, a frame system, a rule-based system etc.

The meaning of a framework is encoded procedurally through the operations available along with the system that operates on the information.

8.5.5 Data Structuring Viewed as a Compression Technique

Here again we will be comparing the methods and techniques in computer systems with Biological information systems. While talking about the concept base, we have already seen that the most basic operation for Biological information systems is pattern matching and the most basic mechanism is categorization. Another technique by which these systems try to optimize their performance is *compression*. The basic technique for compression is what we will call *factorization* i.e.

taking general information out as a *class*, and describing any item as a member of that class having a few *particulars*.

Description can be about a class or an instance of a class. A description has two parts: a part corresponding to things known to the listener (or the population), and an unknown or new thing which is the contribution of that description. Accordingly a description is divided into two parts. subject and predicate. A description about a class can, in general, talk about attributes and relationships important to know for this class. the normal range of values for these attributes; defaults for values of various attributes in absense of specific knowledge etc.

A description about an *instance or member* of the class can partially specify the *known* values or value ranges of some of the attributes or relationships of that member

The members of a class are often grouped together and treated as an aggregate unit, so that each description need not be elaborately specified but instead can be specified as belonging to a class along with values of specific attributes.

A particular structure is selected for the class because of the 'promise' it has. It can deliver goods, it is most economic and efficient for information storage and retrieval.

McCarthy, the inventor of the programming language LISP, has shown that *nested lists* is a structure general enough to act as a basic structure, for any other structure. We, however, do not stop at the basic structure, but identify some higher level structures, most of them having become universal among computer programmers. The structures are *Sequence, Array, Table, Tree and Chain, graph(network) and chunk.*

Let us see how each structure functions:

- **Sequence:** In a sequence, the elements are physically as well as logically close to each other, and arranged in one-dimensional plane. The physical closeness puts a restriction on a sequence. It cannot be easily extended as and when required, as there may not be any scope for extending it further. However, new sequences

can be created by combining existing sequences, or taking partial sequences. Examples of sequences are character strings. Two sequences can be compared as well.

Applicable operations: length, concatenation, substring, comparison.

- Array : In an array, the elements follow an order. There is a mapping or correspondence for the elements with the natural numbers. Thus elements can be referred to first, second and so on. There is a limit on element size as well as on array size. It is possible to directly access n-th element of the array.

Applicable operations: length, 'get n-th', 'put n-th', 'is empty', comparison.

- Table: In an array, one can get the n-th element, but there is no way of searching based on contents of the array-elements. One has to go through the entire array. In a table, it is possible to go both-ways: from contents to the identifier and from the identifier to the contents.

Applicable operations: 'get n-th', 'search for given search condition involving its attribute values'.

- Tree: A tree structure is used for hierarchical organization of data. (Note: Simon[Simon, 1986] defines hierarchy as follows.

He first defines what a partition means!

Partition

Definition: Let E be a finite set of n elements. A partition P of E is a set of K subsets C_i of E such that,

1. The intersection of any pairs is empty
2. The union is equal to E .

Formally,

$$C_i \cap C_j = \emptyset \quad i \neq j$$

$$C_i \cup C_j = E$$

Hierarchies:

Definition: A hierarchy is a set of partition classes constituting a complete chain, including, in particular, the set E itself and the n subsets formed by the elements of E.

The passage from level K to level K + 1 corresponds to combining n classes.

A hierarchy is a subset H of Partition E such that

1. $E \subset H$,

if x and y are elements of E, then $x, y \in H$

2. if h and h' are elements of H, then either $h \cap h' = \emptyset$ or $h \cap h' \neq \emptyset$ in which case either $h \in h'$ or $h' \in h$.)

Tree(Hierarchy) is a way of grouping and subgrouping elements in smaller and smaller groups. In a hierarchy step-wise selection of smaller and smaller subsets meeting some criteria is possible. It is a systematic way of pruning down the search.

Examples: A journal paper is hierarchically organized with sections, subsections, etc. We assume that our knowledge about any subject acquired-formally, is hierarchically organized. In nature, trees are the examples of hierarchical organization.

- Chain: In a chain, a typical element *points* to the next element. Thus, in a chain, elements can be added to or deleted from or modified, at any time. The physical closeness of elements is not required.
- Graph: Graph or network is a very general structure to represent any system, where transitive properties are important.
- Chunk: *Chunk* is any structure consisting of two parts, *title* and *contents*. *Title* serves as a *handle* to pick up *contents*. The chunks can form an hierarchy.

8.5.6 Example of Compressions in a Fact-base

Factorization is one of the compression techniques.

Suppose we have these facts about a student in a college to start with.

Vijay is a student.

Vijay's age is 29.

Vijay's height is 180 cm.

Vijay's weight is 55 kg.

The common factor is Vijay, the subject(topic).

Thus these descriptions can be written as Vijay is a student and
Vijay's attributes are (age is 29, height is 180, weight is 55 kg).

We get a *record* corresponding to facts about Vijay.

Assume further that, there are many other students about whom, the
same information is available.

For example:

Sujatha's attributes are (age is 26, height is 164, weight is 49 kg)

Smita's attributes are (age is 25, height is 150, weight is 45 kg).

Sandhya's attributes are (age is 27, height is 160, weight is 50 kg)

Sheela's attributes are (age is 25, height is 163, weight is 51 kg).

Attribute names are common in each column.

	Identifier	age	height	weight
Vijay (29, 180, 55)	Vijay	29	180	55
Sujatha (26, 164, 49)	Sujatha	26	164	49
Smita (25, 150, 45)	Smita	25	150	45
Sandhya (27, 160, 50)	Sandhya	27	160	50
Sheela (25, 163, 51)	Sheela	25	163	51

Thus we get a table with columns for attributes. This entire table can
be stored as a chunk with a proper title say "Students" in the fact-base.

Hierarchy also can be explained as a factorization technique, applied

stepwise so that at every stage, level, the commonalities are factored out. We can easily show inheritance hierarchy in frames as a factorization technique.

Another kind of compression are by way of providing mechanism for *inferencing* by formally manipulating the structure. For example, inheritance information can be obtained by traversing a tree. Again transitive relationships (paths, connections) can be found by traversing a network.

8.5.7 Associations between Discourse Structures and Data Structures in Computers

It is evident from the discussion so far, how a particular organization is intuitively selected. It is because of the 'promise' it holds. It can deliver goods and it facilitates browsing or querying.

For procedure and methods we use chunks so that they can be picked up by their *titles*. For delimiting a scope by stepwise refinement (sorting etc), we use trees. When we have to access information only sequentially, we can use sequential files. When we have to get attributes of entities, by specifying their identifiers, we will use records. If we have to get entities, by specifying their attribute values as well, we will use tables. For storing linkages, cause-effect etc, we will use linked-list etc.

8.6 Organization and Scope of Queries

Information is useful only if it can be retrieved. To get the right information, the organization of information should be made available to the system. The organization is a kind of compression that delimits the scope of queries. Organization should be such that retrieval of items is possible

- through Identification(Indexing)

- through specification of Partial state or attributes(Content-based searching)
- through domain specification
- by pointing(Visual display of information on the screen)
- by browsing(Going through skeleton descriptors)

Natural language does provide us with all these techniques to identify the referents.

To support our claim, we quote Narasimhan[Narasimhan, 1981]

Language behaviour, like behaviour in the other modalities can be categorized into one of the three meta-behavioural categories, describing, manipulating, and exploring. In the language modality the traditional terms for these categories are, declarative, imperative, and interrogative. These linguistic classifications, however, concern themselves with the surface forms of utterances.

Description: Descriptions could describe the behavioural aspects of the world to which the language relates, i.e., the situational aspects and/or the agentive aspects of either the speaker or the addressee. Describing a situation of the world involves specifying what objects are present, what their states are; what relationships, if any, hold between these states; what events are happening; what relationships hold between these events etc.

A description of agentive aspect could relate to the agent's on-going acts, agentive states, or knowledge, belief, ability, it could relate to the agent's action in terms of its intent; it could relate to a contemplated plan of action by the agent or to the constraints conditioning such a plan (in terms of obligation, compulsion, etc.); it could relate to the agentive states(need, want, desire, etc.). A description could be an assertion or denial.

The primary behavioural intent of a description is to provide information. This is the principal part of the description - thesis part. Aside from this, a description, in general, could also contain a comment-part indicating the speaker's assessment of the validity of the thesis, his degree of belief in it, whether the thesis is being asserted or is being put forward as a hypothesis, the source of knowledge for its declaration, and so forth.

To make communication through language possible at all, the correspondences between the utterances and the behavioural aspects as available at these interfaces should also be, by and large, comparable among all the individuals of a language community.

Thus, among these members the details of their interpretational systems should agree to a large extent. These are just the aspects that show up as the common aspects of the language behaviour of that community. This is just another way of saying that a child builds up his interpretational system by interacting with the language community and, hence, in conformity, by and large, with the interpretational systems of the individuals of that community.

We shall confine ourselves to a small subset of those that relate to situational and agentive aspects that deal with naive language behaviour. Naive language behaviour is what is universal among all human beings

The simplest way of identifying an agent or object is through the use of specific individual names: proper names. Quite often, especially with respect to entities other than human beings, it may be convenient to use class names. We have already referred to the use of personal pronouns as demonstratives. this one, that man, or using observable properties.

Identifying an agent or object - especially an object - in terms of its location is the next most convenient alternative.

Events unfold in time and, hence, specification of time assumes fundamental importance in description of events.

Thus, in descriptions, specification of space and time relationships play an important role. Two other kinds of relationships that play basic role are order relationships and part-whole relationships.

Concerning the states of object, the most important are those directly perceivable by an agent.

As for agents, their agentive states (needs, affects, motivations) and changes in these states are the aspects of primary importance to naive language behaviour.

In English all descriptions are obligatorily assigned explicitly a time relationship.

We have already seen that agents and objects have associated attributes and assigning a value to an attribute determines a partial state of an agent/object.

- 1. attribute and value names explicitly stated.
its colour is blue.
variant
it has a blue colour.*
- 2. Attribute referred to by its associated sensory interface.
it smells sweet
it seems small
partial state is explicitly given
the water is cold*
- 3. An attribute value of an agent or an object can also be specified indirectly by comparing it or equating it with that of another agent or object.*
- 4. Other relationship specifiers:
Just as time relationships arise in relating two events, space relationships arise, in relating two or more agents and objects.
Order relationships could of course, relate either two or more events of two or more agents/objects. Part-whole relationships arise when referring to members of a col-*

lection, or to separable parts, or portions, of an object.
(page 103)

In descriptions of events and states, location in different ways

1. *specifying the place of action*
2. *action involves placement of an object / agent*
3. *place of agent / object when in state described*
 - *demonstrative*
 - *indirect location specifiers*
 - *displacement specifiers*
from - to

Action identification already delimits, more or less in complete detail, the kind of displacement specification called for.

Displacements can be specified through demonstratives.

Names which are values of the attributes length and distance could be used to specify displacements directly.

- *he walked one mile*
- *Agent / object specification*

In descriptions of events, agent specifications are used to identify 1) the primary agent (the one who acts) 2) the secondary agent, if any (the beneficiary / victim), or 3) both. Object specifications are needed to identify the objects manipulated by the agent, or the objects referred to as a pseudo-agent. In state descriptions, these specifications identify the agent / object whose state is being described.

It is possible to specify aspects of an event (or a state) indirectly through reference to other events and / or states. Hence, an event or state description could conceivably refer to a plurality of agents and objects.

Agents / objects can be directly specified either by means of demonstratives or through the use of their proper names.

Agents/objects can be specified indirectly in terms of their attribute values. An agent can be identified in terms of a secondary event.

An object can be specified in terms of its usage, locality.

In propositional speech non-elementary relationships exist between situational aspects. Examples are: Conditionally relating two aspects, causally relating two aspects, implicationally relating two aspects, relating them as hypothesis and consequence, evaluating an aspect etc.

8.7 Conclusion

The world of information consists of various kinds of relationships: time relationships, space relationships, order relationships, and part-whole relationships. Some of them capture the relationships in reality. We feel organization of descriptions is very important. The organization is a kind of "compression" that delimits the scope of queries. Various structures should co-exist and kinds of relationships they capture must be explicitly specified. Data descriptions should be phrases in natural languages, so that the meaning of the descriptors and meaning of the structures impart the meaning to the whole structure.

Justifications for data structures are necessary. The soundness of a particular structure can be verified if only the characteristics of the data to be stored in the structures are made explicit and if only the characteristics and limitations of various data structuring formalisms are known. Thus structures are operations carried on descriptions. Once the structures have "meanings" associated with them, a user interface, in say Singlish for accessing them will be more natural. We will consider this aspect in the next chapter.

Chapter 9

ReWire - Real World Information Retrieval

9.1 Introduction

The point is not just that we can handle large chunks of knowledge as though they were atoms. the important thing is that we should be able to find our way through these complex, nested structures to whatever individual fact or relationship we might need at any given time. We should be doing this in a very flexible and efficient way and we should be avoiding having to look individually at each of the vast number of facts which are not relevant to the problem at hand.

— Sowa[Sowa, 1984]

In previous chapters, we have described a methodology to represent information. However just representation of information is not enough. Information is meaningful only if it can be retrieved. In this chapter, we will discuss the methodology to be used to retrieve information and also about additional mechanisms that are needed to make it possible. It has already been established that organization of information is very

important. Given the requirements of a particular application, the System Analyst comes out with data structures that meet the needs effectively and efficiently. The choice is guided by the practical needs as stated in the Software Requirement Specification. Nevertheless, we should have a theory about why the models we create are valid, why these representations have been constructed and not any other.

As discussed earlier, there are many formalisms available on computers that can be used to represent information. We find Otsu's [Otsu, 1993] comment on these formalisms adequately describing what we have to say.

The frameworks of information processing by modern computers are still not so flexible compared to human flexibility in information processing in the real world where many problems are ill-defined and hard to describe in algorithms. Therefore in order to cope with such real world problems it is essential to pursue the fundamental ways of human-like flexible information processing.

"Real world computing" is becoming a new paradigm of information processing which aims at furnishing the realworld-ness(or flexibility) of human information processing to information systems.

Human beings can be described as "Real world Information Systems". In neuroscience, the basic unit of human memory is described as a *chunk*. While proposing *Production systems* as general models of cognitive architecture [Anderson, 1983],[Newell, 1990], and as a way to represent human expertise in computer programs, the primitive structure assumed is also a *chunk*. We have also named our basic unit for information representation as *chunk*. (Note that, in our case, it is basically a unit of retrieval where *title* serves as a handle to the *body* of a *chunk*. However, so far, we haven't specified what should go in *titles* and what should go in the *body* of the *chunk*. We will attempt to answer this question in this chapter.

Before attempting this, a study of information retrieval in computer systems as well as in biological systems is in order.

9.2 Retrieval in Computer Systems

Many formalisms like Databases, Frames, Semantic networks, First-Order-Predicate-Logic-based formalism, etc. exist for storing and retrieving information on computer. These are based on what Date[Date, 1993] calls 'Type 1' models.

Type 1 data model is a formal system, involving three components, namely a structural component, an integrity component, and a manipulative component. These components can be applied to the problems of any specific enterprise or organization, note carefully, however, that a type 1 data model, in and of itself has nothing to do with any such specific enterprise or organization.

The system based on any such formalisms - say Relational Database System works because it assumes for the designers of databases the following kind of shared knowledge: - Knowledge about concepts such as relation, entity, domain, or attribute ; knowledge about operations such as arithmetic, set, or database operations; and knowledge about the query language.

For the users of a particular application in a relational database, shared knowledge is the names of relations and attributes, underlying domains, integrity and security constraints, etc. This is nothing but the meta-level knowledge for a particular database instance.

One must note that the query in such a formalism is of a very specialized and limited kind though it is made to look like a natural language query by a smart choice of attribute names that correspond to natural language words.

9.3 Retrieval in Biological Information Systems

The *Biological Information Systems* are complex; they contain different kinds of information, each dictating its own way of handling it. The information is added incrementally. This information is often incomplete, imprecise or approximate.

Let us first list the kinds of information these systems have:

9.3.1 Kind of Information Based on Formal Appearance:

Information based on formal appearance can be described as follows:

- **Rules:** Rules are generally used for describing laws of nature, restrictions humans have to follow and generalities or conditions.
- **Records:** Much of the information in files and databases record the apparent state of the real world at some point in time.
- **Procedures:** Descriptions of changes to the state of real world as a result of explicit events or activities are captured through procedures.
- **Messages or transactions:** These are communications from one agent to another.

Descriptive data is useful if it is formalized, standardised and structured.

9.3.2 The kinds of information based on usage

Information can be classified by usage as follows:

- Explanatory information concerned with explaining why a real world situation arose.
- Qualifying and qualitative information which is used to moderate descriptive information from the formal system.
- Patterns and norms which specify how things should be done
- Judgement information based on subjective or intuitive appreciation of a situation.
- Information about various attitudes and power.
- Time and space coordinates of information
- Problem-solving information.

(Note: A structure for understanding may not be adequate as a structure for taking action.)

Let us now see how human beings are able to make use of this information. Here we benefit a lot from the characterization of human problem solving behaviour by Newell and Simon [Newell & Simon, 1972]. We summarize their work here.

9.4 Problem Search and Knowledge Search

According to [Newell & Simon, 1972]

Intelligent behaviour will be deeply involved with the processes that determine what spaces are searched and control search in those spaces.

There are two separate searches going on in intelligence. One is problem search, which is the search of the problem space (search for the information needed). The other is knowledge search, which is the search in the memory of the system for

the knowledge to guide the problem search. This search for knowledge is not always required.

Given a special purpose intelligent system that works on only one task with a general search algorithm (say branch and bound), then only a little knowledge has to be brought to bear (the evaluation function and the logic of branch and bound) and it is used at every state in the space. The procedure can be built right into the structure of the system, so that only problem search occurs. On the other hand, agents that work on a wide range of tasks say - humans - have large bodies of knowledge. When a new task arises, that body of knowledge must be searched for knowledge that is relevant to the task.

Furthermore, in general this search is not a one time retrieval that initializes a small body of knowledge relevant to the task. Some tasks have such a character, but generally at every state in the problem space there is the possibility that some new knowledge, available in mind but not hitherto retrieved has become relevant. Thus, the knowledge search goes on continually and the more problematic the situation, the more continuous is the need for it. Knowledge search occurs in the inner loop of problem search. Knowledge is used to select the operators for the problem search and to evaluate their consequences, hence knowledge is called upon within each state.

Newell's Unified Framework for Cognition assumes the following things about cognition.

Experiments show a holistic change in peoples' skills. Novice and expert differ less in their physical knowledge than in how they access and use it.

They have distinct strategies for selecting potentially relevant principles from long-term memory; and where a novice carries out several distinct steps, the expert gives the result immediately.

Expertise in general, involves automatization of this sort. Information that is consciously accessed while a skill is being learnt comes to be ignored (may even be inaccessible) once the skill has been acquired.

Our initial verbal representation about what to do gradually give way to un verbalized habits of action.

New rules are created by three main processes - Proceduralization, Composition and Tuning.

Proceduralization takes place when a declarative item has been used in a particular way several times.

Newell, Allen [Newell, 1990]

Let us see now how a particular architecture will help in "problem search" as well as "knowledge search". We will be studying the requirement from the competence aspect as well as performance aspect of the system.

9.5 Competence

Competence is the ability to do a particular job.

The problem space provides the opportunity to respond to each situation in terms of the knowledge available at that moment.

[Newell and Simon, 1972]

Let us first assume a simple information retrieval scenario where an answer to a question exists as one of the descriptions in the system. The problem is to find which description corresponds to the question asked. The input descriptor(query) has to be mapped onto an internal description. The simple-minded solution will be syntactically matching

words from the input description to the words in the internal description. However, this is not enough as every word in a question cannot be a *cue*. In general, the matching words in the query provide the *referent* of the query: the topic or subject of the query. Thus they will only give a partial match. The only other thing that can provide *cue* to further matching is what is called *intension* of the query.

We, therefore, make here a very important statement.

In order to have a general-purpose retrieval system, intensions of queries should be matched with intensions of internal data descriptors. Thus it is necessary in a knowledge-based system to capture intensions of internal descriptions at various levels.

9.6 What is Intensional Organization?

The concept of intension dates back to Frege [Frege, 1949] and his distinction between the sense and denotation of an expression in a language. The concept of intension is meant to capture the notion of the sense or idea or meaning of an expression. Montague defines intension of any expression as a function from a set of points of reference (variously called possible worlds or indices) to extensions [Partee, 1976]. Thus the intension of a name is a function which, given any index, picks out some individual as the referent of that name at that index. Similarly the intension of a set picks out some collection of individuals which is the referent of the set-name at each index, and the intension of a formula is a function which, for any index, tells whether the formula is true or false at that index.

A major role of query is to point; the speaker is directing the hearer's attention to some entity.

What descriptions can be used to identify the referent depends on the beliefs of the speaker and hearer, including the speaker's beliefs about the hearer, and so on.

In speaking and listening people make essential use of a great deal

of world knowledge that they 'share' with each other. The question is what kind of 'shared' knowledge do they use, and how?

We have seen earlier that knowledge can be classified as generic and particular knowledge. Generic knowledge is knowledge about kinds of things (about kinds of objects, states, events, processes, methods, etc.), whereas particular knowledge is knowledge about individual or particular things.

Here we make another important statement.

The intensions of data structures should be formed out of the common (generic or public) knowledge to make a query answerable.

9.7 Performance

Let us now discuss the mechanisms towards improving performance. We have seen in previous chapters that one purpose behind structuring the descriptions is to *compress* them into data structures and the technique followed is *factorization*. We now state that

Another purpose behind structuring descriptions is establishing connections. The four mechanisms for establishing connections are levelling, ordering, grouping and pointing.

We will now describe these mechanisms in details.

9.7.1 Levelling

We work on the hypothesis that intelligent systems are built up of multiple levels of systems. Empirically, everything we understand about engineering systems tells us that building up multiple levels is required when the system is complex. Herb Simons [Simon, 1986] in analysis of hierarchy argued that stability dictates that the system has to be hierarchical. According to him (which we do believe) attempts to build

complicated systems without first building stable subassemblies will ultimately fail.

Levels are clearly abstractions, being alternate ways of describing the same system, each level ignoring some of what is specified at the level beneath it.

9.7.2 Partitioning

Partitioning is basically used to divide and subdivide a set into smaller and smaller sets to narrow down the search.

9.7.3 Pointing

It is a mechanism to reach from one item to another item or to get full information from partial information.

9.7.4 Clustering

The descriptions or particulars of a class are often grouped together and treated as an aggregate unit, so that each description need not be elaborately specified but instead can be specified as class + values of specific attributes.

9.8 Meta-knowledge

In the last section we have seen what the two aspects, viz. competence and performance, mean. Now we put forward another important design consideration:

It is not enough to know that *things* are connected, it is also necessary to know what kind of connection it is!

For example, intension of an hierarchy may be to capture *partonomic* relationship, to capture taxonomic relationship, or to capture spatial relationship.

Once structures and their intensions are known, the retrieval operations that correspond to a query expressed at knowledge level (in natural language) can be dynamically composed as and when query comes.

In other words we can say that in order to act intelligently the system should have 'knowledge' about knowledge:*meta-knowledge*. Here we quote Smith's reflection hypothesis[Smith, 1982].

Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits and b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behaviour that manifests that knowledge.

Smith's Reflection hypothesis goes as follows.

In as much as a computational process can be constructed to reason about an external world in virtue of comprising an ingredient process(interpreter) formally manipulating representations of that world, so too a computational process could be made to reason about itself in virtue of comprising an ingredient process(interpreter) formally manipulating representations of its own operations and states.

We will now worry about how to get *intensions* of descriptions as well as *intensions* of structures i.e. *meta-knowledge* when our descriptions are formed using the methodology described in earlier chapters.

9.9 How to Get Intensions of Descriptions

9.9.1 Text-based Systems

In text, titles and sub-titles capture the intensions of the text that follows.

At a higher level of granularity, to make retrieval possible, descriptions in the *skeleton descriptor* can function as *handles* for the text. Thus one of picking information may be using *skeleton descriptors*.

The lowest discourse unit (say corresponding to a paragraph or a description of a phenomenon) is typically made up of many descriptions which may or may not be related to each other. If related, the relationships can be captured as cause-effect, purpose, functionality and so on.

Each description has a purpose to serve. A description generally has two parts: a part corresponding to things known to the listener, and an unknown or new thing which is the contribution of that description (This roughly corresponds to the two parts of a sentence: subject and predicate.)

Therefore a way to make the function of a description explicit is by (1) capturing the subject (topic or theme) of the sentence and (2) capturing the intension (The intension of a sentence may in turn depend upon the kind of the subject.)

Let us illustrate this through a simple story in English.

It was summer (intension: specifies time)
a crow was flying to go somewhere (intension: specifies Action)
the crow was very thirsty (intension: specifies State)
his throat was dry with thirst (intension: specifies characteristics)
he was suffering a lot (intension: specifies state)
he was looking in all the directions (intension: specifies act)
but there was no water (intension: specifies state)
he was tired (intension: specifies state)

he perched on the branch of a tree(intension: specifies act)
he was now thinking(intension: specifies mental act)

The question now is, how to get these intensions from the text. Here we feel the semantics of the verb phrase will help.

Semantic Categorization of Verbs

Very often, it is the verb used that serves as a key to the kind of description.

Let us sub-categorize the verbs based on the purpose they serve.

The three special verbs in English 'be', 'have' and 'do' function as follows:

- to be: to describe a property, identification, class membership, states etc.
- to have: to describe possessions, parts of a thing etc.
- to do: to describe actions.

Different actions, happenings, feeling, effects and states are described using different verbs. For example, [Quirk, 1985] has classified the verbs to denote

- Acts: tap, kick
- Activities: hunt, play
- Events: arrive, die
- Goings-on: rain, snow,
- Accomplishments: discover eat
- Processes: improve, separate
- States: become
- Stances: stand, sit

9.9.2 Fact-based Systems

For fact-based systems, intensions of the structures can be explicitly stored by the author (designer of the application) as meta-knowledge. In a data-structure, intensions can be captured from data descriptors, if data-descriptors are written as noun-phrases using Singlish. In chapter 9, we have described how noun phrases are written in Singlish.

Thus retrieving the name of John's father will depend upon how the information is stored, as a relation or a frame or a semantic network or a record.

9.10 Intensions of Input Descriptors

In English, in input descriptors (questions) wh-words specify what is the unknown that is wanted (intension), while the rest of the description describes what is the known aspect of the description (referent).

In the SQL statement in relational databases,
select name, salary from employee where employee name = "Shobha"
in general, 'select' specifies for what attributes, you want attribute values; 'from' specifies, the class (relation) of interest, while 'where' specifies the restrictions on attribute values.

9.11 Retrieval through Inferences

Up till now, we have assumed a simple case where there is a one to one correspondence between a description and a query. In reality, however, not all references are explicit. Here we quote [Narasimhan, 1981].

It is possible to specify aspects of an event (or a state) indirectly through reference to other events and/or states. Hence,

an event or state description could conceivably refer to a plurality of agents and objects.

Agents/objects can be directly specified either by means of demonstratives or through the use of their proper names.

Agents/objects can be specified indirectly in terms of their attribute values. An agent can be identified in terms of a secondary event.

An object in terms of its usage, locality

In propositional speech non-elementary relationships exist between situational aspects. Examples are Conditionally relating two aspects, causally relating two aspects, implicationaly relating two aspects, relating them as hypothesis and consequence evaluating an aspect etc.

Thus a referent can be specified by specifying

- contextual information
- place
- time
- relationship with other entities

Therefore, we need a mechanism to retrieve information through a variety of means depending upon what user finds convenient at that point of time. We also should provide mechanism to change levels and granularity of the unit under consideration. The organization of information should be transparent to the user.

Data value can be a primitive type or another data element.

While information in elementary data values can be searched only through search and scanning except when it is underlined in which case an index is maintained and it acts as a trigger for the respective data element.

data value:type

Type of the data description is *one* of the primitives of the description language.

9.12 Conclusion

If we want the system to be flexible, retrieval should be possible in many ways: deliberate, through meta-knowledge, multiple scanning and searching.

One important factor that should be taken into account, while describing data descriptors is their semantic content, which can be used to guide queries. In any knowledge representation formalism we observe that data is divided into two kinds - data description and data value. What forms the intension(data descriptor) and what forms an extension(data value) depends upon the semantics of the data as well as the needs of the users. **We take the view that data description is more or less permanent(stable) part of the data structure and data value is a dynamic part.**

The thing to bear in mind is that the intension should provide a 'handle' to lift the data. In other words, intension and extension should correspond to the title and body of a chunk respectively.

Chapter 10

Implementation

10.1 Introduction

An obvious advantage of expressing a theory in the form of a computer program is that it is rendered entirely explicit and its logical coherence is put to a stringent test. If the program works and does what it is intended to do, then the theory it embodies is at least internally consistent.

— [Johnson-Laird & Wason, 1977]

In earlier chapters, we have described the methodology that we should follow. In this chapter, we will be describing the implementation of some of those ideas.

In Appendix X we will be describing CIBA's concept base and its meta-language. We will also be presenting examples of *simple concepts*, *concept mappings* and *concept clusters* in Appendix IV, VII and XI respectively.

Now we will describe *Interpret*: the interpreter for Singlish that converts Singlish sentences into a *canonical form* by making sentential structure explicit, by attaching roles to parts-of-speeches and by linking various clauses and other units. In *Interpret*, two particularly

interesting modules are: the reduction method for part of speech disambiguation and the parser based on Marcus's deterministic parsing. We will also describe two other systems: *Understander* for getting semantic roles of constituents of a sentence and *Skeletonizer*: a system to get skeleton information from the printed text.

10.2 The Interpreter

Syntax is the system of rules that explains how the components of the language are put together to form sentences. Semantics is concerned with the rules for attaching meaning to the sentence of the language.

The structure implicit in the sentences can be made explicit by parsing the sentence. This makes explicit the roles taken by various concepts in a sentence.

Sentences are ways of expressing thoughts and describing various things. Thoughts are composed of concepts. Meaning of a sentence is derived from the constituent words and the structure compositionally.

If the concepts are unique, the meaning of the sentence is also unique if the structure is unique i.e. if there is a single 'parse' for the sentence.

We work on the assumption that

- there is no fundamental difference between a natural language and a formal language when it comes to parsing sentences if we can disambiguate parts-of-speech.
- the primary function of language syntax is to help in conveying the meaning of the sentence.
- the requirement of compositionality can be met if the syntax of a Natural language can be used for semantic compositions in the streamlined language.

Let us now describe the algorithm followed by Interpret. (Please see Appendix XII for various signs and classes used by Interpret.)

10.2.1 The Algorithm

The overall algorithm for interpreting a Singlish (Streamlined) sentence goes as follows:

- Conversion into Singlish (if it is a complex English sentence).
- The separation of adverbial clauses, compound clauses, punctuations, and relative clauses.
- Identification of the kind of sentence (statement, command,...) based on end-markers.
- Expansions of shortforms, apostrophies
- Separation of adverbs in the beginning and in the middle.
- Morphological analysis giving Base word plus features
- Identification of 'candidate' concepts for each word.
- Elimination of some of the 'candidate' concepts by eliminating nongrammatical combinations.
- Building a matrix of candidate concepts associated with all words.
- Building the matrix showing all candidate compositions.
- Separation of the sentences into clauses.
- Use of reduction method to reduce combinations to a unique composition. This involves
 - Identification of the verb (finite or infinite)
 - Identification of foreground elements, based on verb-subcategorization (d_1 -transitive, etc)

- Building of a role-matrix by specifying candidate roles for each word.
- Use of the parser based on Marcus's deterministic parsing to get descriptions into *canonical forms*.

We will now describe, two major procedures, Reduction method to disambiguate part-of-speech and the Parser which is based on Marcus's deterministic parser.

The Reduction method

The principle we follow is in tune with what Johnson-Laird has said

Human reasoners exercise intelligence in ways that are conspicuously absent from machine reasoning. They are able to draw conclusions for themselves, and in so doing they abide by sensible constraints. For instance, they do not normally throw the semantic information away.

[Johnson-Laird, 1983]

The idea is to determine the roles of concepts and identifiers (open class words) in a sentence using closed-class words like conjuncts, determiners, prepositions, pronouns, question words, auxiliary verbs and relative clause words. We take into account the absolute as well as relative positions of words. We also take into account the type of the word (noun, adjective, adverb or verb) The problem arises when the same word is for a noun as well as adjective or noun as well as verb and so on. We consider each such combination, and look for the cues in the surrounding words to get its unique meaning. This goes in parallel, and resolution of ambiguity in one pair helps resolution of ambiguity in the others. For example, if it is a single clause, and the main verb is identified then some other word which can be either noun or verb will be assigned a type noun.

10.2.2 The Parser

To interpret a sentence we first need to *parse* it. The word *parser* comes from the Latin *pars orationis* (part of speech). The basic function of a parser is to determine the function of each word in a sentence - not only its part of speech but also the way it is grouped in phrases with other words

Two most common forms of parsers for context free languages are operator precedence and recursive descent. Operator precedence is especially suitable for parsing expressions, since it can use information about the precedence and associativity of operators to guide the parse. Recursive descent uses a collection of mutually exclusive recursive routines to perform the analysis.

Recursive descent parser has one fatal flaw for natural languages, namely, it cannot parse any language that has ambiguities or non-determinism and therefore is not suitable for natural languages. Marcus's deterministic parser [Marcus, 1980] uses both top-down and bottom-up techniques and with limited look-ahead can parse normal natural language sentences deterministically.

Deterministic Parsing

The most significant development in modular theories of syntactic analysis within the last ten years has been Marcus's (1980) attempt to construct a deterministic model, one that does not rely on arbitrary choice and the indiscriminate use of backtracking that inevitably results.

Marcus postulated a hypothesis that bulk of language syntax processing by humans is done on the basis of commitment to certain readings in one single pass. The hypothesis itself stated that natural language could be parsed by a mechanism that operated in a strictly deterministic fashion. Deterministic parsers (DP) serve to optimize on both space and time by sacrificing some amount of analyticity towards making intelligent decisions in advance with a degree of look-ahead of input and current machine state. Bottom-up parsers start growing their parse trees from

low-level constituents upto the highest levels like noun phrase, verb phrase and finally sentence. With a sufficient look-ahead, a bottom-up parser can be completely deterministic. Marcus implemented PARSIFAL, a bottom-up parser with a buffer that can hold three partially processed phrases. In parsing English, the information needed to make a choice may be arbitrarily many words ahead. But Marcus maintained that for normal sentences, the correct choice could be made by looking no farther than three phrases ahead. In fact PARSIFAL adopts a wait and see attitude, and scans upto three phrases before deciding how to link them together.

Criticism of PARSIFAL

One major criticism about Marcus's theory is that the theory fails to address the issue of genuine structural ambiguity. In particular, he proposed that the syntactic analyser would produce only one analysis of an input sentence at a time, even if others were possible, and that if this analysis proved incorrect, the analyzer would be called on the same input again, with some provision made to block the original, erroneous analysis. The main defect of this approach, however, does not lie in the additional costs, in terms of increased complexity, that are imposed on the syntactic module itself. The real problem here is that the determinism and modularity are preserved in the syntactic analyzer at the expense of non-determinism in the language analysis as a whole.

Data structures for Deterministic parser

Two major data structures of the deterministic parser are

- Push down stack of incomplete constituents
- Three place constituent buffer that the interpreter uses

Pushdown stack of incomplete constituents allows us to deal with the recursive properties of natural language. The parser operates by attempting to add constituents to the node at the bottom of the stack. It

covers up its incomplete constituent with other nodes while it is building the lower level constituents that are its children. When a node is completed, it is popped off the stack and the parser then continues adding constituents to the node which is immediately above it in the stack. Thus, in general, if one node is immediately under another node in the parse tree, it will be immediately under that other node in the active node stack as well.

The primary data type in this parser is the parse node. Grammatical structures are represented by tree structures of parse nodes, each node representing one grammatical constituent. Each node in a completed tree is of a given type and has a parent (except the root), a list of children and an associated set of grammatical features.

The constituent buffer is the central feature of the grammar interpreter that distinguishes this parser from all others - the words that make up the parser's input first come to its attention when they appear at the end of this buffer after morphological analysis. After the parser builds these words into some larger grammatical structure at the bottom of the active node stack, it may then pop the new constituent from the active node stack and insert it into the buffer if the grammatical role of this larger structure is as yet determined. The parser is free to examine the constituents in this buffer, to act upon them and to use the buffer otherwise as a workspace.

In general, the parser uses the buffer in a first-in, first-out fashion. It typically decides what to do with the constituent in the leftmost buffer position only after taking the opportunity to examine its neighbours to the right.

Each cell in the buffer can hold a grammatical constituent of any type, from a single word to a complete sub-ordinate clause. Constituent is any tree that the parser has constructed under a single root node. It must be stressed that the size of the structure underneath the node is immaterial. The input stream provides the parser with lexical items that have undergone morphological processing. From the point of view of grammatical processing, lexical items enter the end of the buffer on demand. After the parser has decided that the current active node is

complete, it is always free to insert that node into the leftmost buffer cell and then pop the node from the active node stack if it does not know what higher level structure the newly completed current active node is attached to.

The key limitation on what the parser can build as a single constituent and then drop into the buffer is not the length of the constituent but rather the fact that there is no facility within the grammar interpreter for backtracking of any kind. Furthermore the only way the parser can construct a constituent is to attach all subconstituents to the topmost node of that constituent. There are no registers or other mechanisms that the parser can use to declare the bounds of a constituent while storing its pieces until the parser is sure of their roles. The parser must be absolutely sure that the structure it is going to assign to a constituent is correct before building it. A further restriction on the parser is that feature assignment as well as node attachment is permanent. Features can be added to a node but not removed.

The ability to buffer constituents potentially gives the parser a great deal of information to use in deciding what higher level role should be assigned to a constituent, but there is no recourse should the parser make a wrong decision.

A Rationale for Two Data Structures

Pushdown stack is a natural data structure for top-down parser. Such a parser is constantly attempting to add sub-constituents to some specific node in the parse tree, which it does by postulating a sub-constituent of that node and recursively repeating the process until it postulates some terminal category that can be checked against the input string. This can be implemented by using a pushdown stack, treating the node most recently pushed onto the stack as the node to which subconstituents must be added. When a node is complete, it can be popped from the stack and attached to its immediate predecessor on the stack, after which the parser automatically continues to add further constituents to its predecessor.

Similarly for a bottom-up parser, the data structures which are most natural are those which allow easy access to contiguous constituents *into a higher level constituent*, repeating the process until a root node has been constructed that incorporates all the terminals in the input string. One of the key operations in such a parser is the process of examining a sequence of contiguous constituents in the course of establishing whether or not they form a higher level constituent. A data structure that allows random access to contiguous constituents is the natural choice for implementing such a process.

A node is pushed onto the active stack whenever the parser seeks to add subconstituents to its internal structure. However, rather than postulating and then attempting to build possible sub-constituents of the current active node in a purely top-down manner, the parser activates a number of pattern-action rules that attempt to recognize and then initiate subconstituents of the current active node by examining contiguous sequence of constituents in the buffer. The activation of rules which recognize appropriate subconstituents for a given node is a top-down process, while the triggering of these rules once activated is bottom-up.

Constituents can also be initiated in a purely bottom up manner by pattern action rules that are always active no matter what the current active node. Such a rule recognizes the presence of a constituent by noting a sequence of buffered sub-constituents that always initiates that constituent regardless of the grammatical environment. Once a constituent is initiated by such a rule, it is pushed onto the active node stack where the parser will then operate in the mixed mode described immediately above, attempting to complete its internal structure by activating pattern-action rules appropriate for that type of node.

The active node stack contains nodes to which the parser is attempting to add children which is an essentially top-down process; the buffer contains nodes for which the parser is attempting to find a father which is essentially a bottom-up process.

10.2.3 Our Approach

If we could eliminate all *Part of Speech* ambiguity before even attempting to structure the sentence, an enormous amount of potential non-determinism is reduced.

It should be quite obvious that relieving the parser kernel of this functionality would ease the burden on it and allow it to be extremely *deterministic*.

Grammar Input Notation

Here we benefit from the work on PASTEL [Srikant & Irani, 1991], the system designed for machine translation at NCST. The grammar input to PASTEL is written in a language similar to PIDGIN which is a formal language that reads much like English, although its syntax is very restrictive and completely artificial. The grammar input is internally converted to LISP by a parser written using *lex* and *yacc*, the two powerful tools available in Unix environment to write parsers, which provide a powerful framework for constructing grammars for simple, token-based deterministic computer languages.

The grammar is written in forms of rules. Rules are grouped in *packets*. A rule can go in one or many packets. Certain conditions activate a packet of rules. Each grammar rule is of the general form

<pattern> → <action>

A rule is active if any packet it is in is active.

In many language parsers there is no distinction between the grammar and the parsing mechanisms, i.e. it makes all the declarative structure of the grammar implicit. The parser is a collection of pattern-action rules, each of which specifies an action to be taken on a set of data structures that are being built during parsing. An action is triggered when a particular configuration is detected by corresponding pattern. Some principles are used to decide the actions to be taken and the control discipline directs the order of application. By separating the

mechanism for parsing from the grammar, we are able to change the grammar of the language, augment the grammar when necessary or even change over to a completely different grammar (say grammar for an altogether different language) by changing the input given to the program that builds the ultimate parser for that language. Some sentences on which we have tried this parser are given in Appendix I. This parser is used to model the functional grammar, which can be used to transform a natural language sentence into a canonical functional form.

Some sample sentences parsed using this parser are listed below:

the president elect may be speaking about his
policies on yesterday ;intransitive
the boy has killed a pig ;monotransitive
the boy killed a pig ; monotransitive
the cruel boy killed a very fat pig yesterday
the girl gave baby a toy ;ditransitive
the girl seems restless ;copular
the girl makes me mad ;complex transitive
he is my brother; copular
the boy who is playing cricket is my brother
;clause with a relational pronoun
playing is fun ;clause with a present participle
the boy playing cricket is my brother; infinite clause with -ing
the boy may have played cricket; complex verb phrase
the boy may have been playing cricket; complex verb phrase
the boy may be playing cricket; complex verb phrase
the boy has played cricket; past perfect tense
the boy has been playing cricket; complex verb phrase
the boy had been playing cricket; complex verb phrase
the boy was playing cricket; past continuous

We have taken sample sentences from English usage, and tried to analyse them using our parallel algorithm. Appendix I gives all the sentences under consideration, Appendix II gives the sentences which could give roles to all constituents uniquely. Appendix III gives the

problematic sentences.

10.3 From Syntactic Roles to Semantic Role: Identification of the Topic and Intention

Syntax is concerned with the structure of the sentence. Semantics is concerned with its meaning. Frege drew a distinction between the "sense" of an expression and its "reference". The "reference" of an expression is what it stands for in the world. The sense of an expression is a part of its meaning, the part that concerns the way in which the expression connects up with its references.

Frege's informal and somewhat obscure notions of sense and reference have been replaced by the concepts of intention and extension in formal semantics. Thus semantic interpretation of a sentence in a limited sense, will be the one that will assign a unique meaning (concepts in our case) to words in a sentence, make the intention of parts-of-speech explicit, and thereby intentions of the sentence explicit. To know the intentions of the parts-of-speech, one has to take into consideration what is called *Language in use* [Narasimhan, 1981]. The factors that help in determining intentions of speech are

- Syntax of the sentence
- Underlying concepts, their category, domain, plane, primitive
- Semantic subcategorization of verbs
- Particular *usage*

Let us illustrate this with some examples.

A syntactic rule says that Subject of the sentence is typically a noun (or np or a nominal clause). It

occurs before the verb. The *subject* is often described as the constituent defining the topic of the sentence - that which the sentence is 'about'. We also know that a sentence follows SVO pattern. Thus, just going by syntax it is very likely that we will assign the syntactic role of subject to the first noun phrase in a sentence. However, take the following sentence:

Today Geeta is very busy.

From the concept base, we can make out that the first word is a noun denoting time. So it is likely to be a peripheral adverb and we will see if there is another noun phrase before the verb that fits in the role of a subject.

Consider another sentence:

It is hard to believe that she passed the examination.

In this sentence, the topic of the sentence is the whole predicate and not the word 'it'.

Similarly, in the sentence

It is raining.

when one is describing only the background (environment: time space or climate) and foreground of the description is nil, 'it' is used as a 'dummy' subject.

The semantic sub-categorization of verbs is also helpful in getting the overall intention of a sentence. In case the verb is primary, or copular, the complementary information can help in arriving at the intention of the sentence.

10.3.1 Viewing Knowledge at Various Levels

A sentence processed using *Interpret* and *Understander* can be viewed at various levels of details.

- as a full sentence in Singlish.

- as a summary description having topic(subject) and intention.
- as a partial description having only foreground information.
- as a minimal description with only head-words of phrases.
- as an ordinary English sentence(with Singlish markers removed).
- as a description in full canonical form with roles of parts-of-speech made explicit.

10.4 Skeletonization

We will now briefly discuss the work done towards getting skeleton descriptors for the text from printed documents. We have implemented a system that presently does only character recognition[Gupta & Irani, 1994]. However, it has all the routines ready to develop what we call a *print analyser*. The intelligent scanner can be used to convert the existing paper text into computer readable text using the *print analysers*. While text formatters use the logical descriptions of the concepts like title, footnote, etc. to convert the text into aesthetically pleasing and functionally organized printed pages, a print analyzer can use the page layout and its knowledge of formatting conventions to separate out the organizational information about the text.

Thus the output of *print analyser* will be the text with titles, subtitles, etc identified. We know that these are nothing but noun phrases in English describing the *intentions* of the following text. Once we have this text ready, converting the noun phrases in canonical form is not difficult as *Interpret* is used for analysing noun phrases with head words, determiners, pre- and post- modifiers and relative clauses.

10.5 Mappings between Structures in Two Planes

So far we have seen the system to interpret a sentence and a system to get summary descriptions of the sentences. Now we will see what facilities are available to manage the fact-base. There are various discourse structures in the discourse plane: story, method, procedure, definition, prototype(class) etc. Also there are structures in the computer plane: trees, frames, rules, lists, records, chunks etc. The discourse structures are ultimately mapped into structures in the computer plane as follows:

- Trees are typically used for representing concept clusters.
- Frames are typically used for representing classes, prototypes.
- Tables are used for storing attribute values of instances of entities.
- Rules are used for relationships like cause-effect, for choosing methods, for selecting steps in problem solving.
- Lists are used for storing sets, or for storing instances of a class.
- Records are used for clustering attributes of instances.
- Chunks are typically used for procedures and methods.
- Links are used for relationships among descriptions: *cause-effect, purpose, functionality, method, manner, convention, belief, definition.*

10.6 The Meta-language for Writing Fact-base

The meta language for writing fact-base will also be Singlish with additional vocabulary in the computer plane. In computer plane as well as in linguistic plane we add a few more primitive classes

- rules
- structures
- categories
- measurements
- relationships
- schedules

The objects in computer plane will be data structures, formalisms, programming languages, and data descriptors. A system can be defined using them. Structures in computer plane are further categorized into chunks, trees, frames, rules, lists, records.

Let us illustrate how we can write meta-language in Singlish:

```

<ID> is a computer system.
Purpose of <ID> is to ...
Domain of <ID> is Commercial.
The programming~language is c++.
The formalism used for DataStructuring is Oracle~version~7
with Forms version 5.0 and ...
The operating system is Unix 5.0 or above.

The method to install the system is a Makefile ``a.txt``.

```

See Appendix XIII for an example of data description using explicit phrases.

10.7 Future work

Getting semantic roles from syntactic roles is not included as part of this study. It needs incorporating a lot of knowledge about English usage (Semantically equivalent to the book [Quirk et al.1985])

Chapter 11

Conclusion

11.1 Summary

To specify a unified theory of cognition is to specify an architecture. An architecture is the system of mechanisms that accesses encoded knowledge about the external world held in memory and brings it to bear to select actions in the service of goals.

If it works perfectly the actions reflect only this external knowledge the system behaves as a knowledge level system.

— Newell, Allen [Newell, 1990]

In this chapter we will be discussing the major thrust of this thesis, summarizing ideas on various issues. We discuss the unique aspects of our approach. We also attempt a brief review of similar efforts elsewhere. We end our thesis presenting some ideas about avenues for future work.

Of late, there is a growing awareness about the shortcomings of earlier knowledge representation systems. On-going in the development of what are called terminological databases is also reported. Various

groups are advocating different formalisms as "best" for the job. Controversies abound in this field. However, most of the theory is not put into practice. Reasons for this, according to us, are

- Many issues are handled in isolation. We have to *Unify* them.
- Methodology and tools are not provided in all the cases.
- An enormous amount of work is needed to build an underlying base.

Some of the suggestions made in the course of this presentation:

- The knowledge representation framework should provide ways and means to the one who designs the knowledge base to state what it is all about in an unambiguous way to enable the system to search relevant information.
- Canonical and unambiguous representation of knowledge with flexible input-output gateways is crucial for the world that hosts as many as 2 languages.
- It is advantageous to take lessons from nature! Mankind's success can be attributed to his invention of language as a medium for representation of knowledge. Language is definitely based on some stable concepts; otherwise, it would have failed.
- Language is not only a medium for external communication, but for internal communication, planning, decision making and problem solving at a conscious level.
- Despite the differences, there is a large degree of commonality among languages, for example the division of words into classes which are syntactically equivalent. We must exploit this commonality.
- Concepts from natural languages should be the building blocks for computer-based systems as well. Concepts are not isolated units,

but inter-related. We should provide enough explicit knowledge about concepts to make them unique. We should also build higher level concepts (terminological bases) in various disciplines on top of this concept-base in a modular fashion. Navigation through these bases will be made possible if we organize them properly.

- Any adequate theory of the human concept system will have to give an account of how concepts are grounded, structured, related to each other and defined.
- Concepts from natural languages should be the building blocks for these systems. We should also take into account the granularity while matching various concepts. Exact match is difficult and in many cases not required. Thus we should quantify the interrelationships among concepts to make approximate match possible. Formalization of commonsense knowledge is a must. And here we should take the help of many existing knowledge sources.

In this thesis we have suggested criteria for selection of an *ideal book* to provide *raw text* as well as *skeleton* for the text on a particular subject. The criteria apply to the books on a given subject with the same *scope* (i.e. number of concepts). The formulae need modifications to include books with different *scopes*. We have assumed that the *best book* on a subject exists, which may not be the case always. Further research is needed in this area if we have to integrate text from different text-books in a single system.

Designing of a knowledge-base for factual information should not be based on adhoc methods. If queries are to be answered at what we have been calling the computational level, some principled way of structuring knowledge must be found as the structure of knowledge has a bearing on what can be answered.

We feel that various structuring methods should co-exist and depending upon the requirement they should be employed. We should understand the "semantics" of each structuring method. We should also view them as structures which do "semantic compression" and in that process provide only particular kind of access to the information. Concepts

from natural languages should be the building blocks for these systems too.

Formalization of commonsense knowledge is a must. And here we should take the help of many existing knowledge sources.

11.2 Evaluation of CIBA

11.2.1 Unique Aspects

The unique features of CIBA are

- CIBA unifies various natural languages
- CIBA unifies various formalisms for representing knowledge
- CIBA unifies syntax and semantics of a language
- No deep knowledge is needed to start using the system
- CIBA keeps idiosyncrasies of a particular language at bay
- CIBA takes into account language in usage.
- CIBA provides a sound base
- CIBA makes semi-automatic knowledge retrieval possible from text-books

11.2.2 CIBA and Semanticity

In this work, we stress semantic organization of information. In Sense and reference in a Psychologically based semantics - Ray Jackend-off[Ray, 1984] has given the essentials for a semantic theory: We summarize it here.

Most theories assume that at least the following four conditions must be met by an adequate semantic theory

First, it must be able to express unambiguously all the semantic distinctions made by a natural language.

Second, in order to account for the fact that languages are (largely) intertranslatable, the stock of semantic expressions available to particular languages must be universal, that is, the semantic well-formedness rules must be universal. (This does not mean that every language is necessarily capable of expressing every possible language).

Third, a semantic theory must provide some principled way for the meanings of the parts of a sentence to be combined to arrive at the meaning of the whole sentence. This requirement of compositionality may be taken more or less strongly, according to whether one requires each constituent (as well as each word) of a sentence to be provided with a well-formed interpretation.

Fourth, a semantic theory should be able to account formally for the so-called "semantic properties" of utterances, such as synonymy, anomaly, analyticity & presupposition. In particular, the notion of valid inference must be explicated.

We also summarize work by Davis [Davis et al, 1993] providing criteria for knowledge representation systems.

Five distinct roles a Knowledge representation plays (Davis et al)

1. knowledge representation is most fundamentally a surrogate, a substitute for the thing itself.
2. It is a set of ontological commitments
3. It is a fragmentary theory of intelligent reasoning expressed in terms of three components
 - a. the representations fundamental conception of intelligent reasoning
 - b. the set of inferences that the representation sanctions
 - c. the set of inferences that the representation recommends

4. It is a medium for pragmatically efficient computation
5. It is a medium of human expression, that is a language in which we say things about the world.

The framework we have suggested is designed taking into consideration these criteria.

11.2.3 Justification for Our Approach

Why a natural language sans its ideosyncrasies is the right choice for computer system representations?

1. Over the years natural language has been used to represent a variety of knowledge.
2. It is a universal convention (within that language speaking community).
3. Already a lot of knowledge is available in natural language form.
4. In other formats, sometimes, there is loss of information.
5. Many times, to select a particular representation, it is essential to know the purpose for which that knowledge will be used.
6. Many times, the formalism requires complete information which is not available.

11.3 Related Work

The strategy of using a calculus to represent and manipulate ideas seems to have originated as far back as seventeenth century with Leibniz. Pioneering work in formalization of natural languages was done

by Chomsky. The concept of grammars as psychological theories had a central place in the thinking that brought about the Chomskyan revolution in linguistics [Chomsky, 1975].

[Sowa, 1984] is an attempt to provide a general, philosophical and psychologically sound foundation for most work in AI. It culminates many years of his work on Conceptual Graphs

Brian Smith [Smith, 1982] has expressed the need for a representation scheme to talk about itself through his *knowledge Representation Hypothesis*.

One of the ongoing pursuits in knowledge representation research involves the role of primitive knowledge structures. The important question discussed is what primitives are appropriate to build into a representation and at what level [Schank and Rieger, 1974], [Brachman, 1979], [Bobrow and Winograd, 1977].

First-order logic has always played a central role in Knowledge Representation research. [McCarthy, 1977] was among the early papers that advocated it for representing and using the commonsense knowledge. [Winograd, 1983] advocated procedural representation of knowledge. KRL, a knowledge representation language was designed to build a system for language understanding. [Minsky, 1981] introduced *frame* as a unit for structuring knowledge. [Newel, 1990] proposes *production systems* as a unified architecture for cognition.

[Schank, 1975], [Wilks, 1987] dealt with the problem of semantic primitives. [Rosch, 1978] talks about the existence of *basic level* in mental organization. Various groups are working on developing *Ontologies* for knowledge representation [Lenat & Guha, 1990], [Nirenburg, 1992]. Work is also on in development of terminological databases. Researchers are using LDOCE - Longman's Dictionary of Contemporary English [Procter, 1978] like dictionaries to extract knowledge [Boguraev & Briscoe, 1988]. Though not in knowledge representation, the importance of Interlingua is very much accepted in machine translation [King, 1987], [Nirenburg, 1987]. Many basic concepts in structuring are formalized in [Simon, 1986].

Readings in Knowledge Representation [Brachman et al, 1985] is a good collection of earlier work in Knowledge representation

Linguists like Lev Vygotsky [Vygotsky, 1986] have dealt with philosophy of language and its relation to thoughts. [Narasimhan, 1981] deals with modelling language behaviour by studying language behaviour of children.

CIBA inherits many ideas from these researchers.

However, CIBA is different from all these because of its emphasis on computational model.

Recently, there is awareness about the shortcomings of earlier knowledge representation systems. Work is also on, on development of what is called terminological databases. This work is similar to our work in spirit.

11.3.1 Japanese Real World Computing Program

A new 10 year project launched by Japan in 1993 called "Real World Computing Program" with a budget of over \$ 500 million over a period of ten years, essentially aims at the same theme: at laying theoretical foundation and pursuing the technological realization of human-like flexible and intelligent information processing as a new paradigm of information processing towards the highly information-based society of the 21st century.

Statement

"Supported by the remarkable development of computer and communication technologies, information technology is producing an innovative change in the society, not only in industrial activities but also in the qualitative improvement of our way of life. It is foreseen that information to be handled will explosively increase toward the next century because of increasing needs of multimedia information processing and the expansion of new application domains. It means not only the increase in quantity but also the increase in quality and variety of information

Such social and technological needs are starting to require a new paradigm of information technology, not as a simple linear extension of the conventional one, but as an essentially new underlying framework. In other words, it is necessary to make computers more friendly and easy to use by providing them with human-like flexible and intelligent capabilities in order to assist and collaborate with humans in the diverse information environment of the real world.

The framework of information processing by modern computers is still not so flexible compared to human flexibility in information processing in the "real" world where many problems are ill-defined and hard to describe in algorithms.

Therefore, in order to cope with such real world problems and to open a new horizon in information processing technology, it is essential to pursue the fundamental ways of human-like flexible information processing, by casting light on the intuitive or "subsymbolic" level of human information processing.

"Real world computing" is proposed as a new paradigm of information processing which aims at furnishing the realworldness (or flexibility) of human information processing to information systems.

In the traditional information processing, humans (users) are forced to adapt themselves to accessing computers using hard logic. In future, computers will be expected to get close to humans and support human intellectual activities, collaborating in the diverse information environment of the real world.

It will also be important to learn and get inspiration from nature, namely to take into account new findings in scientific research into the brain, evolution process of creatures, and ecological dynamic systems.

Merely combining conventional technologies or to make ad-hoc systems for specified tasks is not what is desired."

11.4 Contributions of this Research

1. Singlish is very much like English, thus reading or writing in Singlish does not add to cognitive load and therefore is easy for human beings.
2. In computer science, very little published work is available on semantic classifications. Our cross-linguistic study has helped us in getting many 'semantic relationships'. We have built groups of *basic words*, *primitive hierarchies*, and *verb clusters* which we think are useful for future work.
3. We have attempted to remove the surface language barrier from the knowledge-driven systems. Idiosyncrasies of surface structures of a language are removed to some extent and therefore many interfaces are possible. Various other gateways like *Shindi*, *Smarathi*, *Stamil*, *Smalayalam* can be made available to make the concept-base available to a larger community.
4. Building a *concept-base* is not a Herculean task needing hundreds of man-years. We already have around 3000 *simple concepts*. One immediate application we can think of is getting interpretations for *concepts* from dictionaries by reading a dictionary using OCR.
5. The grammar of English that the parser can use can be easily extended. In another project at NCST, [Srikant & Irani, 1991], we have shown how language usage can be taken into account and incorporated into system as 'data' which the parser can use in turn.
6. Conversion into skeletons helps in viewing text at various levels of details.
7. The skeletonization also helps in browsing. Thus one can enter a suitable environment and then work in a *limited context*.
8. By providing support for automatic acquisition of information from printed text-books, the knowledge acquisition bottleneck is removed.

9. Searching by establishing context and then using descriptions optimises search which can become very time-consuming otherwise.
10. The meta-knowledge is easily available. The overall organization of the information system built on the top of this formalism can be easily seen in human-readable form. This helps in knowledge acquisition, representation and retrieval.
11. Information can be incrementally added to a system based on this formalism, as everything is 'transparent'.
12. Alternate hierarchies can be provided for the information easily, as *chunks* can be picked up by their *titles*.
13. Data modelling based on concepts for which there are words in natural languages, makes knowledge explicit, unambiguous and universal.
14. The most important thing is that we have put theory into practice. We have implemented a rudimentary translation system from English to Hindi on this concept-base that works!

11.5 Shortfalls

We haven't said much about any computational method for matching. We feel that *Semantic distances* among concepts will play an important role towards approximate matching. In [Srikant & Irani, 1991] we have described formulae for semantic distances between concepts. However, we strongly feel that much more work needs to be done in this area.

With current limitations of OCR technology, scanning books using OCR is still not as easy as reading floppies. Thus getting knowledge from text-books may not be immediately feasible.

11.6 Future Work

Human common-sense reasoning is influenced by incomplete information. Except for the most trivial cases, there are always gaps in our knowledge. Yet, we are not paralyzed by our partial ignorance. We are skilled in reaching rational conclusions. Computer systems should also be made to work with incomplete information.

Everything cannot be meaningfully expressed in logical language. Real world knowledge is rarely of this categorical form.

Probability can be regarded as more fundamental than logic, because logic is just a special case within probability theory (where probabilities are only 0 or 1), but the converse is not true.

Probability in some form is a necessary component of practical reasoning. One goal, therefore, must be to find how to make probability practical.

The main thrust of this thesis is on the importance of building, modifying, verifying, extending etc., models albeit imperfect in some sense.

We feel institutionalizing language like Singlish is very important. Recently, we had experimented with a notation Interscript [Irani & Ram, 1992] for writing Devanagari using English alphabets. Initially, it was just a coding scheme devised during development of the software Rupanthar. However, it is now found so convenient that it has been used as an intermediate notation and at times even for inputting and editing Devanagari.

Conventionally, learning a script of a language seemed to be a precondition to the learning of the language itself. By adopting an encoding scheme like Interscript, one of the major barriers is removed and consequently rapid progress can be made in language learning at both the grammatical and conceptual levels.

Our system CIBA does enforce some new conventions. However, when the characteristics of a communication system change, some new conventions are inevitable. This has been a regular feature in the past.

When communication system changed from gesture-based to spoken language, from spoken to written, from written to printed, and from printed to computer-based, new conventions had to be introduced.

Computer science is a field mature enough now to have a 'suburb' (to use the term popularized by Wittgenstein [Wittgenstein, 1953]) of a natural language. Mathematics has its language, law has its language. In Singlish we see the emergence of Computerse.

List of Appendices

Appendix I	: Sentences Selected for Parsing using Interpreter	1
Appendix II	: Illustrative Subset of Appendix I	4
Appendix III	: English sentences Problematic for the Current Implementation	6
Appendix IV	: Examples from the Dictionary of Simple Concepts	7
Appendix V	: The words that can act as Primitive or Basic	8-B
Appendix VI	: Illustrative Subset of Domains	11
Appendix VII	: Illustration of Mappings in CIBA	12
Appendix VIII	: Snapshots of the Parser Output	13
Appendix IX	: PASTEL: A Parser for English	19
Appendix X	: The Meta-language	21
Appendix XI	: Illustrative Concept Clusters in CIBA	24
Appendix XII	: Symbols used by Interpreter	26
Appendix XIII	: Example of an Explicit Data Description	29
Appendix XIV	: Two-Phase-Matching Protocol	32
Appendix XV	: Descriptions in Conceptual Information Base 'CIBA'	34

Appendix I

Sentences Selected for Parsing using Interpreter

George made it clear that he disapproved the idea.
James said that he was feeling better.
Mary needs a friend with whom to play.
actually, I can't come.
any boy clever at games should know this rule.
cricket is not terribly interesting to watch.
he came into the room.
he is collecting money for the blind.
he is looking for a place in which to live.
he said she was sorry to have missed you.
he seemed to be smoking a lot.
he seems to have been sitting there all day.
he was to have been the new ambassador.
I agree about politics.
I agree on a date.
I agree to a suggestion.
I agree with you.
I am coming soon.
I am glad that you are all right.
I am looking for something to clean the carpet with.
I am the oldest in my family.
I appear to have made a small mistake.
I came to see you so that you would know the whole truth.
I can't see clearly without my glasses.
I didn't expect to be invited.
I don't dance much now but I used to a lot.
I don't know whether to answer his letter.
I find it difficult to talk to you about anything serious.
I gave her a comic to read.
I had an invitation from the people that live next door.
I have got a headache.

I have never known him to pay for a drink.
I heard a strange noise.
I heard some people passing in the street.
I look forward to hearing from you.
I need a box to hold my specs.
I once studied the guitar for three years.
I really must have my watch repaired.
I saw him three days ago.
I tasted the soup suspiciously.
I think it important that we should keep calm.
I thought it peculiar that she hadn't written.
I want to travel.
I was asleep from three to six..
I was never happy at home.
I went there because I wanted to.
I will phone you after I arrive.
I would like to have a sister.
I would like to really understand xyz.
I would rather go alone.
in my opinion the rent is too high.
it is difficult to understand what she's talking about.
it is getting dark.
it is going to rain.
it is good to have finished work for the day.
it is important for the accounts to be ready by Friday.
it is lovely to have people smile at you in the street.
it is nice to be sitting here with you.
it is not a bad place to live in.
it is not much use my buying Salmon.
it is too heavy for you to lift.
it was difficult to sell my car.
my brother got a job to earn money for his family.
my friend is old.
my mother is getting too old to travel.
my old friend told me to go there.
nothing seems to have been forgotten.
relativity theory isn't easy to understand.

see if you can jump across the stream.
she fell unconscious on the floor.
she is easy to get on with.
she is nice.
she is often late.
she is very nice to talk to.
she never listens to the advice which I give to her.
she ought to be told about it.
she sang beautifully.
that is the doctor who lives next door to us.
the British are very proud of their sense of humour.
the engine is very quiet.
the house where I live is very small.
the next house to the royal hotel is mine.
the plane was flying over Bombay.
the success obtained in the first six months was great.
the temperature is three degrees above zero.
the valley lay quiet and peaceful.
the water came up over our knees.
there is a big black cat in the bathroom.
there is a lot of work to do.
there was a girl water-skiing on the lake.
unemployment is a current problem.
we are about to have lunch.
we are meeting next Tuesday.
we have always lived in this house.
we have got to get up at six tomorrow.
we live by the side of the sea.
we walked across the road.
we will soon have your car going again.
when I went home, it was late.
you always misunderstand me.
you are completely out of your mind.
you can't make an omelette without breaking eggs.
you should check the oil before starting the car.
you would better see what she wants.

Appendix II

Illustrative Subset of Appendix I

George made it clear that he disapproved the idea.
cricket is not terribly interesting to watch.
he said she was sorry to have missed you.
he seemed to be smoking a lot.
he was to have been the new ambassador.
I am glad that you are all right.
I appear to have made a small mistake.
I came to see you so that you would know the whole truth.
I don't dance much now but I used to a lot.
I don't know whether to answer his letter.
I find it difficult to talk to you about anything serious.
I gave her a comic to read.
I had an invitation from the people that live next door.
I have never known him to pay for a drink.
I heard some people passing in the street.
I need a box to hold my specs.
I really must have my watch repaired.
I think it important that we should keep calm.
I thought it peculiar that she hadn't written.
I want to travel.
I went there because I wanted to.
I would like to have a sister.
I would like to really understand xyz.
I would rather go alone.
it is difficult to understand what she's talking about.
it is good to have finished work for the day.
it is important for the accounts to be ready by Friday.
it is lovely to have people smile at you in the street.
it is nice to be sitting here with you.
it is not a bad place to live in.
it is not much use my buying Salmon.

it was difficult to sell my car.
my brother got a job to earn money for his family.
my mother is getting too old to travel.
nothing seems to have been forgotten.
relativity theory isn't easy to understand.
she is easy to get on with.
she is very nice to talk to.
she never listens to the advice which I give to her.
she ought to be told about it.
that is the doctor who lives next door to us.
there is a big black cat in the bathroom.
there is a lot of work to do.
there was a girl water-skiing on the lake.
we are about to have lunch.
you can't make an omlette without breaking eggs.
~~you~~ *should* check the oil before starting the car.
~~you~~ *should* see *what she wants.*
you would better see *what she wants.*

Appendix IIII

Sentences Problematic for the Current Implementation

James said that he was feeling better.

Problem because better is both a modifier and an adverb.

Mary needs a friend with whom to play.

Problem because of the 'whom + infinite clause'.

he is looking for a place in which to live.

Problem because of the compound phrase 'in which'.

I will phone you after I arrive.

Problem because after is both a preposition and a conjunct.

I am looking for something to clean the carpet with.

Problem because of the deferred preposition.

I look forward to hearing from you -

Problem because of the lack of knowledge of 'phrasal verbs'.

we will soon have your car going again.

Problem because 'object' of the sentence is a compound object.

Appendix IV

Examples from the dictionary of Simple Concepts:

(Note: We can build task-specific dictionaries on the top of CIBA by including extra fields. For example, we have included the corresponding words (in Rupanthar code [Irani & Ram, 1992]) from Marathi and Hindi in the dictionary.)

```
-----  
account ((account n ( (ma . <Hisoba>) (hi . <HisAba>) (pr . )) ( |  
  (account v ( (hi . <HisAba|raKanA>) (pr . ABSTRACT|LIT)) ((tense  
add ((add v mt ( (ma . <joDaNe/adhika|karaNe>) (hi . <joDanA/adhi|  
  (pr . )) ( (tense . inf) )))  
address ((address n ( (ma . <pattA>) (hi . <patA>) (pr . ABSTRACT|  
  (address v ( (hi . <nivedana|karaNA>) (pr . )) ((tense . inf) )))  
advance ((advance n ( (ma . <karja>) (hi . <karja>) (pr . MONEY|  
  (advance v ( (hi . <Age|baDhanA>) (pr . )) ((tense . inf) )))  
answer ((answer n ( (ma . <javAba/uttara>) (hi . <javAba/uttara>|  
  (answer v ( (hi . <javAba|denA/uttara|denA>) (pr . ABSTRACT|LIT|  
appeal ((appeal n ( (ma . <prArthanA>) (hi . <prArthanA>) (pr . )|  
  (appeal v ( (hi . <prArthanA|karaNA>) (pr . )) ((tense . inf) )))  
back ((back n hd ( (ma . <pATha/pRXTha>) (hi . <pITha/pRXTha>) (pr . )|  
  (back prep ( (hi . <pICE>) (pr . )) ( dummy .junk )))  
bear ((bear n hd ( (ma . <asvata>| (hi . <BAU>) (pr . CREATURE)|  
  (bear v ( (hi . <saHana>) (pr . )) ((tense . inf) )))  
-----
```

Appendix V - PRIMITIVES

(Words that can be taken as Primitives in CIBA)

PROPERTY:

PROPERTY ATTRIBUTE FEATURE DEGREE HABIT SENSE BALANCE SIGHT SMELL
TOUCH GRAIN RANGE HABIT STYLE WILL CHARACTER QUALITY MOTION COLOUR
SHAPE SIZE NUMBER MEASURE

INFORMATION:

INFORMATION MODEL SIGN METHOD PATTERN MESSAGE IMAGE

OBJECT:

OBJECT PLANT THING CREATURE MATTER BODY INSTRUMENT SUBSTANCE
MATERIAL WOOD METAL WATER CHEMICAL STONE AIR SOIL PAPER CLOTH
TOOL MACHINE APPARATUS VESSEL WEAPON STRING

ACT:

ACT EFFECT INTEND INFORM CAUSE DO HAPPEN LEARN JUDGE BECOME
POSSESS WORK MAKE USE CHANGE FORM MOVE EXPRESS THINK ATTEND
KNOW FEEL

ENTITY:

ENTITY ELEMENT SYSTEM ENERGY PRODUCT MIND SELF NATURE
ORGANIZATION UNION UNIVERSE SOCIETY EARTH INDUSTRY MILITARY

STATE:

STATE AFFAIR RELATION FEELING
BELIEF EMOTION IMAGINATION CONDITION
LIFE ORDER LIMIT

HAPPENING:

HAPPENING EVENT EXPERIENCE GAME CONVERSATION CEREMONY MEETING
BATTLE SHOW CRIME

THEME

THEME PROFESSION TRADE MEDICINE POLITICS STANDARD SCIENCE
MYSTERY PRINCIPLE RELIGION LAW VALUE ART LITERATURE SPORTS

Appendix V (Continued)

The words that can act as Primitive or Basic

accept act add admire agree allow ask attack bathe bear beat beg
begin behave blow borrow break breathe build bury buy call can-
cel carry catch change cheat choose clap climb close collect come
compare complain confirm confuse congratulate connect contra-
dict control convince cook copy cover create crush cry cut dance do
draw eat escape fall fear feel fight fill finish float fly get give go
grow guide happening hear help hit hold join keep kill laugh learn
leave lose love make marry move order pay play praise pray put
receive ride rise see sell send show shut sing sit sleep spend stand
steal stop strike subtract swim take talk teach tell tie travel try
understand use vessel wait walk want wash wear win work

Figure : All these words are *actions*

air army art bazaar boundary brand breadth business capital
cargo chain champion charity class colour commerce company con-
tract country courage craft crime custom danger direction edge
electricity family fire game gas gift god government group heat
hole kind law light love manners material measure mind money
music name nature navy noise ornament part piece play poison
power prison shop sound speech success taste trouble war wish
work world

Figure : All these words are *entities*

Appendix V (Continued)

aeroplane aircraft arrow bag bed bird biscuit boat body book bottle
bow box boy brass bread brick bridge brush building butter button
buy cage calendar candle cane cap car carpet chair chalk cheese
child choose cloth coffee container dish dust egg farm flag flour
food fruit glass grass gun insect iron knife land leather liquid
machine meat metal milk oil paper pole prison rice room rope sea
seed ship tea train tree wall water wheel wire wood wool

Figure : All these words are *objects*

able active another bad beautiful before behind below big bright
broad centre character charm chief claim clear clever close cold
colour common complete correct cruel dead different difficult dirty
easy empty end enough fierce flat forward front funny good hard
head healthy heavy here hide high hollow holy ill large left less
little long low magic middle more narrow near open part pleas-
ant plenty polite poor present proper quick ready real rich right
right/left round safe same sharp shift short side skilful small sour
start steady straight strong sudden sweet tall there thick warm
wise with wrong young

Figure : All these words are *properties*

Appendix V (Continued)

accent advice agree amount answer any can cause cent century certificate chance chapter character chart comma concerning count coupon everyone everything letters like line list map marks news number phrase plan purpose question rank report say sentence someone something story think word write

Figure : Words in Discourse Plane

Appendix VI

Illustrative Subset of Domains

Generalities Science & Knowledge Organization Information Philosophy Psychology Religion Social science Law Mathematics Physics Chemistry Biology Medicine Arts Recreation Entertainment Sport Language Linguistics Literature Geography Biography History Engineering Agriculture Management Housekeeping Industry & Trade Communication Commerce Military Government Transport Health Vehicle Power & Energy Geography History

Figure 0.1: The *Domains*

Appendix VII

Illustration of Mappings in CIBA

(Note: * indicates that it is the root word)

noun	adjective	adverb	verb
success	successful	successfully	succeed
marriage	married	-	*marry
*hope	hopeful	hopefully	hope
want	wanted	-	*want
*wish	wishful	-	wish
fall	falling	-	*fall

Apart from the standard mappings, there exists some suffixes that can be used for mapping words.

- verb to noun - ing
- adjective to adverb - ly
- adjective to noun - ness
- noun to adjective - ful
- verb to adjective -ing, -ed
- noun to adverb -wise

Appendix VIII

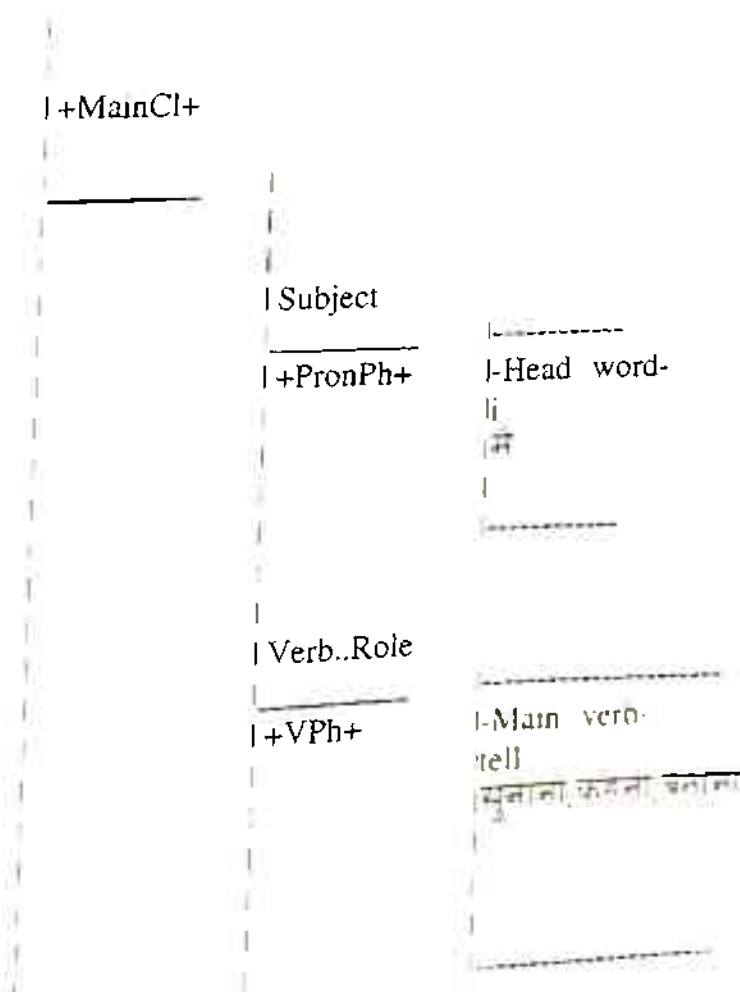
Example of a tree showing description generated by Interpret.

i told the smart girl whom i liked the story.

```

*****
.....
+MainCl+
.....
***** +MainCl+
.....
+PronPh+ +VPh+ +NPh+ +RelCl+ +NPh+
.....
***** +RelCl+
.....
Sub..Con..Phrase +PronPh+ +VPh+
.....

```



| Object..Indirect

| +NPh+

| -Determiner-

| the

| वह

| -Pre Mod-

| smart

| चतुर, साहजिक, जानब

| -Head word-

| girl

| लडकी

| +RelCl+

| Post_mod.Object I.

| Conjunct..Role

| Sub..Con

| -Head word-

| whom

| किसका, किस, जिस

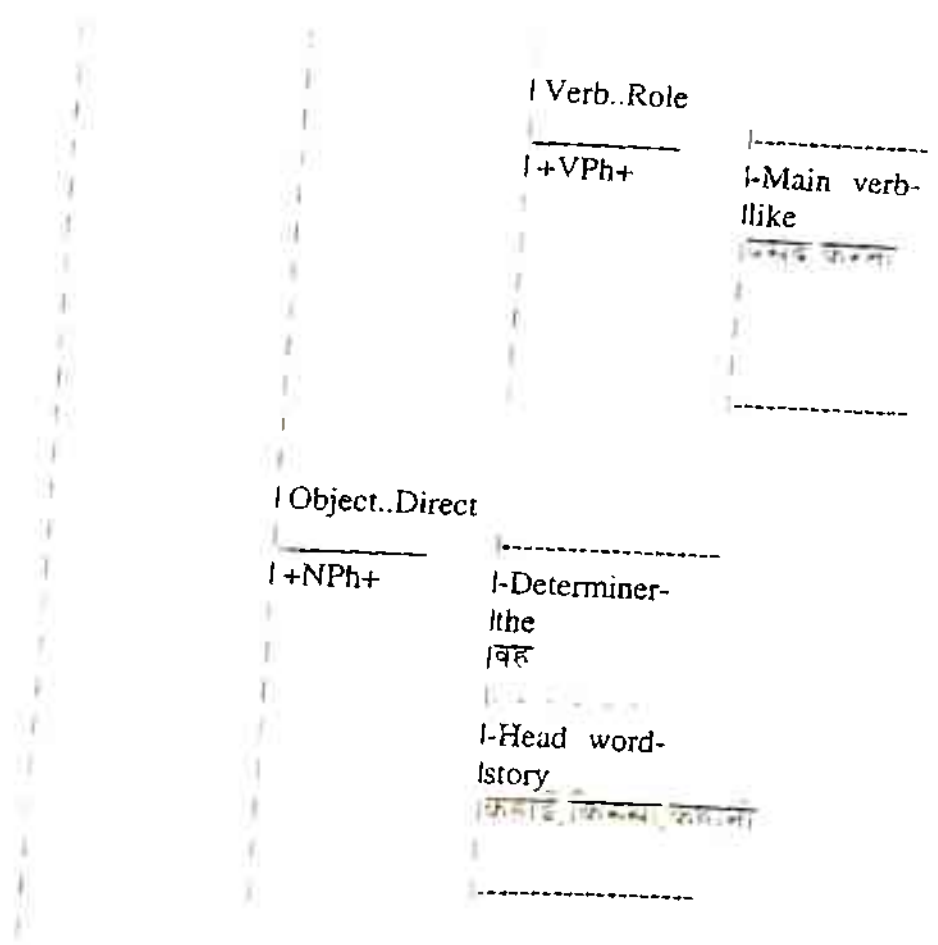
| Subject

| +PronPh+

| -Head word-

| I

| मैं



SYNTACTIC CATEGORIES AND ROLES ASSIGNED BY INTERPRET

i told the smart girl whom i liked the story

Pronoun Fin Verbi Det Adjective Noun Sub Conl Pronoun Fin Verbi Det Noun

Level 1

+PronPh+ +VPh+ +NPh+ +RelCl+ +NPh+
 Subject Verb Role Object Indirect Post_mod.ObjectI Object Direct

Level 2: +RelCl+

+SubCon+ +PronPh+ +VPh+
 Conjunct Role Subject Verb Role

Appendix IX

PASTEL: A Parser for English

(Note: This describes the methodology followed in PASTEL[Srikant & Irani, 1991])

Conventional non-deterministic parsers for natural languages do not clearly delineate separate mechanisms for Part-of-speech (POS) analysis and structural analysis, but instead utilize a uniform principle of failure, to apply any phrase structure rule (i.e. misapplication of grammar rules).

If we could eliminate all POS ambiguity before even attempting to structure the sentence, an enormous amount of potential nondeterminism is reduced. It should be quite obvious that relieving the parser kernel of this functionality would ease the burden on it and allow it be extremely *deterministic*. We try to remove POS ambiguity by three methods: Analytical prepass, statistical prepass and verb subcategorization.

Analytical prepass exploits the idea of making explicit all local (pairs and triples) constraints over syntactic categories to eliminate several invalid combinations.

Since we have already eliminated certain combinations by analytic restrictions, and since our final goal is to obtain a sequence of unambiguous syntactic categories (before structuring), it is obvious that remaining multiple combinations in each span have to be selected for by some means. Here we adopt corpus-based statistical data to choose the best possible arc in each span.

It might be apparent that, at this point, all categories except verbs (noun, adjectives, adverbs and functional words) have been disambiguated upto their subcategory grain. Each verb is augmented by all its possible verb subcategories, in order to choose one of them.

A comprehensive syntactic analysis seeks to identify constituent and dependency structure of sentences. In a functional constituent grammar(FCG), every clause is analyzed by assigning its constituents and subconstituents to structural functions(which are local and syntactic,describing relationships among structures). Such a description lies in between the pure Functional and Immediate Constituent Phrase structured grammars. On one hand it is able to achieve just the adequate embedded structure required from bottom-up analysis, and on the other hand it preserves the intuitive universality of the grammatical relations. In this scheme of things, syntactic categories and their subcategorization play the roles of lowest level of subconstituent.

A situation-action(procedural) grammar(SAG) is written as a collection of situation-action rules each of which specifies an action to be taken on a set of structures that are being built up in the course of parsing. The action is triggered by some particular configuration of structures being built. A cataloging scheme and a control discipline is maintained.

Appendix X

The Meta-language

In chapter 5, we have described how a simple concept can be represented in CIBA using dimensions: primitive, basic, category, plane and domain. We have also described what each dimension means. Here we will describe what are the constituents of each category.

Primitives:

- Act
- Agent
- Object
- Entity
- State
- Happening
- Theme(Subject)
- Information
- Time
- Space
- Property
- Quality

Important Categories:

- noun
- pronoun
- adjective
- adverb
- verb
- conjunct
- preposition

Important Functional roles of the words in a sentence

- Noun
- Modifier
- Adverb
- Conjunct
- Determiner
- Preposition
- Object
- Possesive
- That
- Verb to
- Verb do
- Verb have

- Auxiliary modal
- Auxiliary not
- Full verb
- Question word

Planes

- Physical plane
- Mental plane
- Discourse Plane
- Computer Plane

Appendix XI

Illustrative Concept Clusters in CIBA

GAME - chess, football, cricket, crossword, puzzle

SHAPE - square, circle, rectangle, hole, corner, curve, triangle

BODY - head, limb, hand, leg, stomach, abdomen, finger, toe, eye, ear, nose, mouth, tooth, tongue, palm, chest, thigh, chin, shoulder, hair, ankle, beard, arms, back

RELIGION - christian, hindu, muslim,

PLACES- city, country, world, region, town, continent, village, state, county

ANIMALS - BIRDS - crow, cuckoo, cock, hen, duck,
ant, spider, butterfly

INSECTS - cockroach, ant, spider, butterfly

WATER ANIMALS - cod, crocodile, lion, pig, donkey,

MAMMAL - cow, buffalo, horse, tiger, lion, pig, donkey, rat, dog, cattle, sheep, camel, bull, deer

RELATION - mother, father, son, daughter, sister, brother, wife, husband, grandmother, grandfather, uncle, auntie, mother-in-law, father-in-law, brother-in-law, sister-in-law

MONTHS - Jan, ... Dec

DAY - Sunday, Monday, ... Saturday

COLOUR - red, blue, yellow, orange, black, white, green, brown

NATURE - CLIMATE - rain, cloud, cold, cool

SEASON - monsoon, summer, spring, autumn, winter, fall

LAND - cliff, mountain,

WATER - sea, lake, river, coast, creek, beach, shore, bay

PLANT - PART - stem, root, flower, leaf, fruit, branch, bud

TYPE - bush, creeper, climber,

MATERIAL - cloth, cotton, wool, silk, nylon, coal, coke, cement,
sand, concrete

METAL - gold, silver, steel, iron, copper, zinc,

HOUSE - (see the list)

SOCIAL - club, society, committee, colony, crowd, community,
conference, council

PARTS - coil, stick, column, cord, cork

EDUCATION - school, college, curtain

FURNITURE - table, chair, seat, bench, bed

HEALTH - ILLNESS - chill, fever, typhoid, cough, ..

medicine, clinic, doctor, pathology

exercise

vitamin

ENTERTAINMENT - drama, cinema, circus, play,

CLOTHES - garments, shirt, trousers, blouse, sari, skirt

EDIBLES - VEGETABLE - potato, onion, cabbage,

FRUIT - apple, mango, banana, cherry.

SNACKS - tea, coffee, bread, cake, pastry, chocolate,

icecream, custard, cocoa, cream

Appendix XII

Symbols used by Interpreter

Punctuations Signs

- dot
- comma
- colon
- semicolon
- question-mark
- single-quote
- double-quote
- hyphen
- exclamation-mark
- dash
- opening-bracket
- closing-bracket

Single Punctuation and other Signs

- tilde
- backslash
- double backslash
- asterix

- arrow
- curly brackets
- square brackets
- double exclamation marks
- back-quotes
- empty determiner

Closed-class Words

- determiners
- prepositions
- pronouns
- conjuncts
- modal verbs
- primary verbs

Verb . Subcategorization Information

- intransitive
- monotransitive
- copular
- ditransitive
- complex-transitive

Tenses

- Aspects - progressive, perfect, passive
- Tenses - present, past, future
- Modalities -

Information Associated with Nouns

- Gender - masculine, feminine, neuter
- Number - singular, plural

Information Associated with Adjective - Degree

- Ordinary
- Comparative
- Superlative

Information Associated with Adverbs - subtypes -

- Conjuncts
- Adjuncts
- Subjuncts
- Disjuncts

Appendix XIII

Example of an explicit data descriptiono

(Note: The words in Capitals are the *Concepts* from the Computer Plane while words in square brackets are concepts from the discourse plane.)

[library management] IS THE THEME
[reader] IS AN OBJECT
DESCRIPTION OF [reader] IS ``[reader] IS A [Borrower]``
ATTRIBUTES ARE
[name of the reader] : (NAME PERSON)
[address of the reader] : (PLACE PERSON)
[city in which the reader lives] : (PLACE PERSON)
[maximum number of books the reader can borrow] : (NUMBER PERSON)
[priority of the reader] : (SCALE PERSON)
[money deposited by the reader] : (MONEY PERSON)
[category of the reader] : (SCALE PERSON)

[book] IS AN OBJECT
ATTRIBUTES ARE
[name of the book] : (NAME BOOK)
[edition of the book] : (SCALE BOOK)
[name of the author of the book] : (NAME PERSON)
[name of the author of the book] : (IDENTIFICATION BOOK)
[accession number of the book] : (NUMBER BOOK)
[price of the book] : (MONEY BOOK)
[category of the book] : (SCALE BOOK)
[category of the book] : (NAME COMPANY BOOK)
[name of the publisher of the book] : (NAME COMPANY BOOK)
[agent for buying the book] : (NUMBER BOOK)
[number of pages in the book] : (NUMBER BOOK)

[borrow] IS AN ACT
DESCRIPTION OF [borrow] IS [reader borrows a book]
OBJECTS ASSOCIATED ARE [reader book]
ATTRIBUTES ARE
[name of the reader] : (NAME PERSON)

[name of the book] : (NAME BOOK)
[accession number of the book] : (IDENTIFICATION BOOK)
[date of borrowing the book] : (DATE BOOK)

[recommend] IS AN ACT
DESCRIPTION OF [recommend] IS [reader recommends a book]
OBJECTS ASSOCIATED ARE (reader book)
ATTRIBUTES ARE
[name of the reader] : (NAME PERSON)
[name of the book] : (NAME BOOK)

Let us now see how the system will try to guess the 'most appropriate data element' for the following query.

Who has written the book 'Algorithms and Complexity'?

This question can be changed into a *canonical description*, using Interpreter.

Processing the query - Phase I

The query contains the name of a BOOK and a PERSON is to be searched *in connection* with the BOOK. The act mentioned in the question is *WRITE*.

From the descriptions in data model we gather the following:

- There are four descriptions in the data model, two describing OBJECTS and two describing ACTs.
- The concept 'WRITE' or its synonym is nowhere in the head of any of the data descriptions.
- A structure (NAME PERSON) appears in all the four descriptions.
- A structure (NAME BOOK) appears in second, third and fourth description.
- Thus second, third and fourth descriptions have (NAME PERSON) as well as (name book) specified in them.

Processing the Query - Phase II

The problem now is to get the person associated with the *BOOK*. The modifiers or ACTs associated with (NAME PERSON) should be such that they match the the concept *WRITE* as closely as possible.

Thus further analysis of the phrases is required.

In second relation, the PERSON is (author of the book). In third relation, the PERSON is involved in the ACT of borrowing. In the fourth relation, the PERSON is involved in the ACT of recommending. If we compare the semantic distances between the *concepts* (WRITE and AUTHOR), (WRITE and BORROW) and (WRITE and RECOMMEND) we will find that (WRITE and AUTHOR) are the closest. Thus (author of the book) in the second data description is the preferred data element for the query.

Appendix XIV

Two-Phase-Matching Protocol

Semantic distances between concepts

We have built a discriminational ontology which divides the semantic primitives in eleven basic equivalent classes. Each class has a tree structure assigned to it. We also have a dictionary of *concepts* with each *concepts* defined using semantic primitives of the language. For data modelling purpose, we extend the basic equivalent class to include *number, scale, name, identifier, money and date* as basic categories. We call this class *extended equivalent class*. The semantic distances between any two *concepts*¹ can be calculated using the method described in [Srikant & Irani, 1991]. Our word sense definitions are structured in such a way that an algorithm can most efficiently compare or match two such structures.

The matching technique utilizes a quantitative measure to evaluate the closeness of structures. Our definitions of *concepts* consist of two parts: the first part corresponds to vertical generalization i.e. the category to which the *concept* belongs called the head of the concept. The second part called the tail-set of a concept is a list of features with features appearing in decreasing order of importance. The semantic distance between any two *concepts* depends upon the head of the concept as well as the tail-set of the concept. A measure of semantic distance between *concepts* is derived as follows: If the heads of the *concepts* match exactly then it is a trivial case. Otherwise the closeness of one concept with the other is calculated over the entire ontology. The parameters involved are the distance, depth and height in the ontological trees to which the heads belong.

Following points are taken into account while deriving the formula.

¹This describes the work which was not done as a part of this study. Major work is due to M Srikant who worked in KBCS group at NCST.

- More specific the common ancestor, closer are the two *concepts*.
- More skewed the path pair, closer are the two *concepts*.
- Lesser the conceptual distance, closer are the two *concepts*.

Two phase matching protocol using semantic distances between phrases

Given an explicit data model similar to one defined below, with the help of CIBA and its environment, the system is able to get for each phrase describing an attribute, its extended equivalent class and its most salient features in decreasing order of importance.

Semantic distances among phrases are composed out of semantic distances among the *concepts* of the phrases. A typical noun phrase has a head word and a few modifiers. A new structure can be formed by merging the modifiers with the extended equivalent class of the head word.

Appendix XV

Descriptions in Conceptual Information Base 'CIBA'

A *description* exists on a *computer plane*. It corresponds to a sentence in the *discourse plane*. It is a *relation* that exists in the *computer plane*. It is a unit of information. It describes a situation in which a number of participating *concepts* are involved.

Various *concept expressions* take *roles* in this relation. The *roles* they take are *governed* by a particular kind of concept. (We call it *verb* in *discourse plane*.) *Concepts* are also categorized so that they can take only a certain kind of *role*. The *categories* (they correspond to the *grammatical categories* in the *discourse plane*) are *noun*, *adjective*, *adverb*, *verb* etc.

The *roles* (they correspond to parts-of-speech in *discourse plane*) are *subject*, *direct-object*, *indirect-object*, *verb*, *adverb*, *complement* etc. The *verb* governs the other *roles*.

Verbs can be divided on the basis of how many *concept expressions* are necessary to make them meaningful.

Functions of Major Categories of Concepts

The primary function of the *concepts* belonging to various *categories* in a description are as follows.

<u>function</u>	<u>Type</u>
Naming	Nouns and pronouns
predicating(stating or asserting)	verbs
modifying	adjectives, adverbs
connecting	prepositions, conjunctions

Simple Descriptions

Using the *concepts* belonging to different *categories* and *primitive classes*, we can form simple *descriptions*. Though *descriptions* are divided into four major syntactic types, viz. *declarative, interrogative, imperative and exclamative*, we will be dealing with only *declarative descriptions* in this study. *Description* can be *simple* or *multiple*. *Multiple description* consists of *clauses*. A *description* can be summarized as a pair: *topic* and *aspect*.

Every *description* has a *topic* in general. (Generally *subject* of the corresponding sentence in *discourse plane* serves this purpose. The subject typically specifies what the sentence is about.)

Aspect is what the *description* is trying to say about the *topic*. It can be an attribute, state, act, happening or relationship. (It typically corresponds to the *predicate* of the corresponding sentence in *discourse plane*. It describes 'what is asserted about the subject'.

A simple *description* chiefly involves the elements having *syntactic roles* - *subject, verb, object, complement and adverbial*.

Verb is the most central role in a *description*. It is easier to identify and it determines what other elements may or must occur in the clause.

Subject of the *description* is typically a noun (or np or a nominal). It

is obligatory (except in imperatives where it is implied). It determines *number* and *person* of the verb. *Subject* can be considered as the *topic* of the *description*. It typically refers to information that is regarded by the *speaker* as given.

Objects can be *direct* and *indirect*. *Objects* are normally nouns (or a nominal clause). In *descriptions*, *Objects* follow *subject* and *verb*. If both *objects* are present, *indirect object* comes before the *direct object*.

Adverbs refer to the circumstances of the situation. They come either in the beginning or at the end, or just before the verb.

Complementation is a function of a *concept expression* or a *description* which follows *subject*, *verb* and *objects*(if any), and completes the specification of a meaning relationship.

Thus the five *roles*(functional categories) of description constituents are

- subject (S)
- verb (V)
- object (O) - direct object (Od) and indirect object (Oi)
- complement (C) - subject complement (Cs) and object complement (Co)
- adverbial (A) - subject related (As) and object related (Ao)

The optional elements form the *background* of the *description*. By eliminating the optional adverbs which form the *background* of the *description*, seven **major description types** are established based on the permissible combinations of the seven functional categories.

Major description types are

- SV - intransitive
- SVO - monotransitive
- SVC - copular

- SVA - copular
- SVOO - ditransitive
- SVOC - complex transitive
- SVOA - complex transitive

(Note: The terminology is taken from [Quirk, 1985].

The *description* types are determined by the verb class(*subcategory*) to which the word belongs. Different *verb classes* require either different complementation (Od, Oi, Cs, Co, A) to complete the meaning of the verb or no complementation.

The syntactic *roles* the *concepts* take into a *description* are *subject, object, verb, adverb, complement*.

It is not very difficult to get the *syntactic roles* of the *concepts* if we make the *description* unambiguous.

A typical *description* is formed using *concepts, identifiers, numerals, concept operators* and syntactic markers (punctuations) from *discourse plane* in addition to syntactic markers of Singlish.

References

- [Anderson, 1983] Anderson John R. (1983) *Rules of the Mind*, Lawrence Erlbaum Associates, Hillsdale, New Jersey
- [Arbib, 1992] Arbib, Michael, *From Neurons to Minds Via Schemas: Achieving Artificial Intelligence Through Cooperative Computation.* in *Minds, Brains & Computers: Perspectives in Cognitive Science and Artificial Intelligence.* Morelli et al(eds.), Ablex Publishing Corpo, Norwood, New Jersey.
- [Bobrow & Winograd, 1977] Bobrow D. G. and Winograd T. (1977) *An overview of KRL, A Knowledge representation Language* Cognitive Science 1(1), 1977, 3-46.
- [Boguraev & Briscoe, 1988] Boguraev B K, Briscoe E J (1988) *Large lexicons for natural language processing: Exploiting the grammar coding system of LDOCE* Computational Linguistics 13.
- [Brachman, 1979] Brachman R. J. (1979) *On the Epistemological Status of Semantic Networks* In *Associative Networks: Representation and Use of Knowledge by Computers.* N.V.Findler(ed), Academic Press, New York, (3-50pg)
- [Chomsky, 1975] Chomsky, Noam (1975) *Logical Structure of Linguistic Theory* University of Chicago Press
- [Date, 1986] Date, C. J. (1986) *Relational Database Writings* Addison Wesley.
- [Date, 1990] Date, C. J. (1990) *"Why Relational" in Relational Database Writings 1985-1989* Addison Wesley.
- [Date, 1993] Date, C. J. with Huge Darwen (1993) *Relational Database Writings 1989-1991* Addison Wesley.
- [Davis et al. 1993] Davis, Randall, Shrobe, Howard and Szolovits, Peter (1993) *What is a Knowledge Representation* AI Magazine, Spring 1993

- [Fillmore, 1968] Fillmore C. (1968) *The case for case in Universals in Linguistic Theory*, Bach E and Harms RT (Eds.), Holt, 1968
- [Frege, 1949] Frege, G. (1949) *On sense and nominatum* in *Readings in Philosophical Analysis*, H. Feigl & W Sellars(Eds), pp 85-102, New York: Appleton-Century-Crofts.
- [Grosz, 1986] Grosz, Barbara J. (ed) (1986) *Readings in Cognitive Science* Morgan Kaufman Publishers
- [Gupta & Irani, 1994] Gupta, Ajay G & Irani, Alka (1994) *On the Extraction of features using Mathematical Morphology for document Recognition* Internal memo, NCST available on request to alka@saathi.ncst.ernet.in
- [Jean-Louise, 1990] Jean-Louis Gasse (1990) *The Evolution of Thinking Tools in the Art of Human-computer Interface Design*, Brenda Laurel(ed) Addison Wesley.
- [Johnson-Laird & Wason, 1977] P. N. Johnson-Laird & P. C. Wason, (1977) *Thinking, Readings in Cognitive Science* Cambridge University Press.
- [Hillis, 1985] Hillis, W. Daniel (1985) *The Connection Machine* The MIT Press, Cambridge.
- [Hirst, 1987] Hirst, G. (1987) *Semantic interpretation and the resolution of ambiguity* *Studies in Natural language processing*, Cambridge Univ. Press.
- [Irani, 1990] Irani, Alka *Machine Readable Dictionaries* An internal Memo, NCST, 1990.
- [Irani, 1995] Irani, Alka (1995) *Documentation on CIBA Interpreter* An internal Memo, NCST, 1995.
- [Irani & Ram, 1992] Irani Alka & Sylvia Candelaria de Ram (1992) *Interscript - A "Script" for Computation and Communication of Indian Languages across Contexts* in National seminar on Information Technology in India Languages, Bhubaneswar, India.

- [Israel, 1983] Israel D. J. (1983) *The role of Logic in Knowledge Representation* IEEE Computer 16(10), 1983, 37-42.
- [Johnson-Laird, 1993] Johnson-Laird, Philip N., (1993) *Human and Machine Thinking*, Lawrence Erlbaum Associates, Hillsdale, New Jersey
- [King, 1987] King, Margaret (Ed.) (1987) *Machine Translation: The State of Art*, Edinburgh University Press
- [Lenat & Guha, 1990] Lenat, D.B. and Guha (1990) *Building Large Knowledge-based systems: Representation and Inference in CYC Project* Addison Wesley
- [Levesque, 1985] Levesque, Hector J. and Brachman, Ronald J. (1985) *A fundamental Tradeoff in Knowledge Representation and Reasoning (Revised Version)* in Readings in Knowledge Representation edited by Ronald J Brachman and Hector J levesque, Morgan Kaufmann Publishers, Inc
- [Marcus, 1980] Marcus, M.P. (1980) *Theory of syntactic recognition for natural language* MIT press.
- [McCarthy, 1977] McCarthy, John (1977) *Epistemological Problems of Artificial Intelligence* in Readings in Knowledge Representation edited by Ronald J Brachman and Hector J levesque, (1985) Morgan Kaufmann Publishers, Inc.
- [McCarthy, 1971] McCarthy, John in *ACM Turing award lectures: First Twenty years, 1966-85* Association for computing machinery.
- [McClelland, 1992] McClelland, James L. (1992) *Can Connectionist Models Discover the Structure of Natural Language?* in *Minds, Brains & Computers: Perspectives in Cognitive Science and Artificial Intelligence*. Morelli et al(eds.), Ablex Publishing Corpo, Norwood, New Jersey.
- [Miller, 1956] Miller, G. A. (1956), *The magical number seven, plus or minus two. some limits on our capacity to process information*, *Psychological Review*, 63, pp. 81-97.

haviour Springer-Verlag New York.

- [Narsimhan, 1993] Narsimhan R. (1993) *Relevance of Wittgenstein to Language Behaviour Modelling* Internal Memo, CMC Limited, Bangalore, India.
- [Newell, 1982] Newell, Allen (1982) *The knowledge level* Artificial Intelligence. 18(1):87-127.
- [Newell, 1990] Newell, Allen (1990) *Unified Theories of Cognition* Harward University Press
- [Newell & Simon, 1972] Newell, Allen & Herbert A. Simon. (1972) *Human Problem Solving* Englewood Cliffs, N.J. Prentice-Hall, 1972.
- [Newell & Simon, 1976] Newell, Allen & Herbert A. Simon. (1976) *Computer Science as Empirical Inquiry* Communications of the Association for Computing Machinery 19:113-126.
- [Nirenburg, 1987] Nirenburg, Sergei (Ed.) (1987) *Machine Translation: Theoretical and Methodological Issues* Cambridge University Press, 1987
- [Nirenburg, 1992] Nirenburg, Sergei (1992) *Machine Translation: A Knowledge Based Approach* Morgan Kaufmann Publishers, Inc, Cal.
- [Otsu, 1993] Otsu, Nobuyuki (1993) *Toward Flexible Intelligence: MITI's New Program of Real World Computing*
- [Partee, 1976] Partee, Barbara H., (ed) (1976) *Montague Gram-mers* Academic Press, New York.
- [Procter, 1978] Procter, Paul (ed.), (1978). *Longman Dictionary of Contemporary English(LDOCE)* Longman.

- [Putnam, 1988] Putnam, Hilary (1988) *representation and Reality* The MIT Press.
- [Pylyshyn, 1984] Pylyshyn, Zenon W. (1984) *Computation and Cognition, Toward a Foundation for Cognitive Science* The MIT Press
- [Quirk et al, 1985] Quirk, R. Greenbaum S., Leech G., Svartvik J. (1985). *A comprehensive grammar of the English language* Longman.
- [Ralph et al, 1992] Ralph, Morelli et al(ed.) *Minds, Brains, and Computers: Perspectives in Cognitive Science and Artificial Intelligence* Ablex Publishing Corpo, Norwood, New Jersey.
- [Reichgelt, 1991] Reichgelt, Han (1991) *Knowledge Representation: An AI Perspective* Ablex Publishing Corpo, Norwood, New Jersey.
- [Rieger, 1976] Rieger, C. (1976) *Viewing Parsing as word sense discrimination*, In A survey of linguistic Science, Dingwall W.O. (ed.) Greylock publishers, CT.
- [Rosch, 1978] Rosch, E. (1978). *Principles of Categorization* In *Cognition and Categorization* , Rosch, E. and Lloyd B.B. (eds.) Lawrence Erlbaum, N.J.
- [Ray, 1984] Ray, Jackendoff (1984) *Talking minds* Bever Thomas G. (Ed.) The MIT press
- [Schank, 1972] Schank, R.C. (1972). *Conceptual dependency: A theory of natural language of understanding* *Cognitive psychology* 3, No. 4, 552 - 630.
- [Schank, 1975] Schank, R. C. (ed.) (1975) *Conceptual information processing*. North Holland, Amsterdam.
- [Schank, 1991] Schank, R.C. (1991) *Where's The AI* in AI magazine, Vol.12, No.4.

- [Schank & Rieger, 1974] Schank, R. C. and Rieger, C. J. (1974) *Inference and the Computer Understanding of Natural Language* Artificial Intelligence 5 (4), 1974, 373-412
- [Shannon & Weaver, 1949] Shannon, Claude E. & Warren Weaver (1949) *The mathematical theory of Computation* Champaign, Ill:Univ. of Illionois Press
- [Simon, 1986] Simon, Jean-Claude (1986) *Patterns and Operators: Foundations of Data Representation* North Oxford Academic
- [Smith, 1982] Smith, Brian C. (1982) *Prologue to "Reflection and Semantics in a Procedural Language"* Reading in knowledge Representation, Brachman & Levesque (eds.) Morgan Kaufmann Publishers, Inc, Cal.
- [Sowa, 1984] Sowa, J.F. (1984) *Conceptual structures. Information processing in mind and machine* Addison Wesley.
- [Srikant & Irani, 1990] Srikant M. and Irani, Alka (1990) *A framework for Preference Analysis in Large Scale NLP tasks* Internal memo, NCST available on request to alka@saathi.ncst.ernet.in
- [Vygotsky, 1986] Vygotsky, Lev (1986) *Thought and language* translated and edited by Alex Kozulin, the MIT press, Cambridge, Massachusetts, London, England
- [Wilks, 1987] Wilks, Y (1987). *Primitives*. In *Encyclopedia of Artificial Intelligence*. Shapiro S.C. (ed.) Wiley Interscience.
- [Winograd, 1983] Winograd, T. (1983) *Language as a cognitive process*, Vol I: Syntax. Addison - Wesley
- [Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical Investigations* Basil Blackwell, Oxford.
- [Woods, 1977] Woods, W. A. (1977) *A personal View of Natural Language Understanding* SIGART Newsletter, Feb 1977,(61), 17-20

Publications:

- *Selection Criteria for the Text in Text-Based Systems*, Alka Irani, in Symposium on Document Analysis and Information Retrieval, Las Vegas, U.S.A., March 1992.
- *Adaptive Knowledge Acquisition from real world systems*, Alka Irani in AAAI spring symposium on Cognitive Aspects of Knowledge Acquisition, Stanford Univ., March 1992.
- *Texts Across Contexts: Indic encoding and its applications*, Sylvia Candelana de Ram (Comp. research lab. New Mexico State Univ.), Alka Irani and M Srikant in ACH at Fordham Univ in Bronx, NYC, June 1990
- *Interscript - A "script" for Computation and Communication of Indian Languages across Contexts*, Alka Irani and Sylvia Candelana de Ram (Comp. research lab. New Mexico State Univ.), in Akshara - national seminar on information technology application in Indian Languages at Bhuvaneshvar, India.
- *Experiments with a Knowledge Based System*, Alka Irani, Jitendra Loyal, Sanjay Pathak in Cologne Computer conference, W Germany, Sept 1988
- *Archie - A File Archival System*, Alka Irani, T M Vijayaraman, V Kamala, S M Desai in conference FST & TCS, Bangalore, Dec 1983
- *Design of Software for Text Composition* S P Mudur, A W Narwekar, Abha Mohitra in journal Software Practice and Experience, Vol 9, 313-323(1979)
- *Design and Analysis of Hyphenation Procedure* Abha Mohitra, S P Mudur, A W Narwekar in journal Software Practice and Experience, Vol 9, 324-337(1979)