

# **Sign Language Automation**

*Thesis submitted in partial fulfillment of the requirements for the award  
of degree of*

**Master of Engineering**  
in  
**Software Engineering**

*Submitted By*

**Rupinder Kaur**  
**(801231024)**

Under the supervision of:

**Dr. Parteek Kumar**  
Assistant Professor



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT  
THAPAR UNIVERSITY  
PATIALA – 147004  
**July 2014**

## Certificate

I hereby certify that the work which is being presented in the thesis entitled, "**Sign Language Automation**", in partial fulfillment of the requirements for the award of degree of Master of Engineering in **Software Engineering** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Parteek Kumar** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.



(Rupinder Kaur)

Roll No. 801231024

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

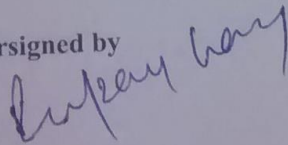


(Dr. Parteek Kumar)

Assistant Professor,

Computer Science and Engineering Department

Countersigned by



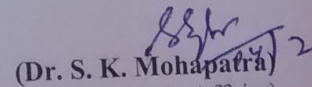
(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University

Patiala



(Dr. S. K. Mohapatra)

Dean (Academic Affairs)

Thapar University

Patiala

## Acknowledgement

---

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds.

With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. Parteek Kumar**, Assistant Professor, Computer Science and Engineering Department, Thapar University for his positive attitude, excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and above all his blessings. He has been a source of inspiration for me.

I am grateful to **Dr. Deepak Garg**, Head of Department, and **Dr. Ashutosh Mishra**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for the motivation and inspiration for the completion of this thesis.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.

(Rupinder Kaur)

Roll No. 801231024

## Abstract

---

Sign language is a visual gesture language. Deaf people use sign language as their main source for communication. They use their hands, face and arms for conveying thoughts and meanings. Sign language is not having any written form and generating signs using video approach is very expensive, so, sign animation is best solution. For performing animation from text or speech, machine translation is used. Indian Sign Language is a linguistically under-investigated language. Research on Indian Sign Language linguistics also limited because of lack of Indian Sign Language knowledge and the unavailability of such tools which provide any education for Indian Sign Language.

In this thesis, sign language generation system for Punjabi language is proposed. It takes Punjabi sentence as input, process it and generates sign animation corresponding to sentence. For generating animation, it parses the sentence; extracts root words from the sentence and generate HamNoSys corresponding to root words. Then HamNoSys is converted into SiGML and animation is generated. HamNoSys generation system is developed in this thesis. It takes word as input and generates HamNoSys corresponding to inputted word. Generated HamNoSys can be saved into its database and used by sign language generation system for Punjabi language.

# Table of Contents

---

---

<b>Certificate.....</b>	<b>i</b>
<b>Acknowledgement.....</b>	<b>ii</b>
<b>Abstract.....</b>	<b>iii</b>
<b>Table of Contents.....</b>	<b>iv</b>
<b>List of Figures.....</b>	<b>vii</b>
<b>List of Tables.....</b>	<b>x</b>
<b>Chapter 1: Introduction.....</b>	<b>1-10</b>
1.1 Sign Language.....	1
1.2 Indian Sign Language .....	1
1.3 ISL Phonology.....	2
1.4 Challenges in Processing of Sign Language.....	4
1.5 HamNoSys: A Notation System for Sign Language.....	5
1.6 Importance of HamNoSys in Generation of Sign Language.....	6
1.7 SiGML Language.....	8
1.8 Avatars.....	9
1.9 Thesis Outline.....	10
<b>Chapter 2: Literature Review.....</b>	<b>11-27</b>
2.1 Direct Translation Systems .....	12
2.2 Transfer Systems.....	12
2.3 Interlingual Systems.....	12
2.4 Existing Systems on Sign Language.....	12
2.4.1 TESSA .....	13
2.4.2 SignSynth Project .....	14
2.4.3 TEAM .....	15
2.4.4 The ASL Workbench .....	15
2.4.5 ViSiCAST Translator .....	16
2.4.6 ZARDOZ .....	17
2.4.7 Machine Translation System from Text-To-Indian Sign Language.....	18

2.5 Available tools on Sign Language .....	20
2.5.1 A Multilingual Multimedia Indian Sign Language Dictionary Tool.....	20
2.5.2 Multimedia ASL Dictionary Tool.....	21
2.5.3 Computer Generated Models for the World Wide Web.....	21
2.5.4 Indian Sign Languages using Flex Sensor Glove.....	22
2.5.5 eSIGNEditor: An Editor of HamNoSys.....	23
2.5.6 HamNoSys System for German Sign Language.....	25
<b>Chapter 3: Problem Statement.....</b>	<b>28-30</b>
3.1 Objectives.....	29
3.2 Methodology.....	29
<b>Chapter 4: Implementation.....</b>	<b>31-54</b>
4.1 Architecture of Proposed Punjabi to Sign Language Generation System.....	31
4.1.1 Tokenizer.....	32
4.1.2 Morph Analyzer.....	33
4.1.3 Part-of-Speech Tagger.....	34
4.1.4 Chunker.....	35
4.2 Description of Proposed System with an Example Sentence.....	36
4.3 Need for Creation of HamNoSys Generation System.....	40
4.4 HamNoSys Generation System.....	40
4.5 Implementation of HamNoSys Generation System.....	40
4.5.1 Representation of Hand Shapes.....	41
4.5.2 Representation of Hand Orientation.....	43
4.5.3 Representation of Hand Locations.....	44
4.5.3.1 Representation of HEAD locations.....	45
4.5.3.2 Representation of Locations related to MOUTH.....	46
4.5.3.3 Representation of PAIRY LOCATIONS.....	47
4.5.3.4 Representation of TRUNK locations.....	48
4.5.3.5 Representation of Hand positions with respect to the body.....	49
4.5.4 Representation of Hand Movements.....	50
4.6 Representation of System by Generating HamNoSys for word “TWO”.....	52

<b>Chapter 5: Results and Discussions.....</b>	<b>55-62</b>
5.1 Testing of Sign Language generation system for Punjabi language.....	55
5.2 Testing of HamNoSys Generation System.....	59
<b>Chapter 6: Conclusion and Future Scope.....</b>	<b>64-65</b>
6.1 Conclusion.....	64
6.2 Limitations and Future Scope.....	65
<b>References.....</b>	<b>66-69</b>
<b>List of Publications.....</b>	<b>70</b>

## List of Figures

---

Figure 1.1 Types of Indian signs.....	2
Figure 1.2 (a) Shows one handed static manual sign of Ear.....	3
Figure 1.2 (b) Shows non-manual sign of headache.....	3
Figure 1.3 (a) Both the hands are moving for signing "long".....	3
Figure 1.3 (b) The dominant right hand is moving for signing "flag".....	3
Figure 1.4 Structure of HamNoSys.....	6
Figure 1.5 HamNoSys symbols and there descriptions.....	7
Figure 1.6 HamNoSys of "WOMAN".....	8
Figure 1.7 Sign of "WOMAN".....	8
Figure 1.8 SiGML for word "DEAF".....	9
Figure 1.9 Different Avatars with their names.....	9
Figure 2.1 Traditional Machine Translation pyramid.....	11
Figure 2.2 Structure of TESSA system.....	13
Figure 2.3 Architecture of SignSynth system.....	14
Figure 2.4 Architecture of ASL workbench.....	16
Figure 2.5 Architecture of ViSiCAST system.....	17
Figure 2.6 Architecture of ZARDOZ system.....	18
Figure 2.7 Architecture of Text-To-ISL machine translation system.....	19
Figure 2.8 Architecture of Indian Sign Language dictionary tool.....	20
Figure 2.9 Flowchart of flex sensor hand glove.....	22
Figure 2.10 eSIGNEditor interface.....	23
Figure 2.11 Searching word in eSIGNEditor.....	24
Figure 2.12 Document window of eSIGNEditor.....	24
Figure 2.13 Hand shapes.....	25
Figure 2.14 Hand orientation.....	26
Figure 2.15 Hand movements.....	26
Figure 2.16 Example of hand movement in "HOUSE".....	26
Figure 2.17 Hand actions.....	27

Figure 3.1 Use of HamNoSys generation systems in sign language generation	30
System for Punjabi language.....	31
Figure 4.1 Architecture of sign language generation system for Punjabi language	36
Figure 4.2 Output file generated by Punjabi shallow parser.....	37
Figure 4.3 Extracted root words.....	38
Figure 4.4 HamNoSys generated by eSIGNEditor.....	38
Figure 4.5 HamNoSys of word “LITTLE” ...	39
Figure 4.6 SiGML for word “LITTLE” generated by eSIGNEditor.....	39
Figure 4.7 Sign animation generated by proposed system.....	41
Figure 4.8 (a) Represents sign for word “EIGHT”.....	41
Figure 4.8 (b) Represents HamNoSys for word “EIGHT”.....	42
Figure 4.9 Hand shapes for generating HamNoSys.....	42
Figure 4.10 (a) Represent sign for word “EIGHT” .....	
Figure 4.10 (b) Represents HamNoSys for word “EIGHT” after selecting	42
Hand shapes.....	43
Figure 4.11 Hand orientation for generating HamNoSys.....	44
Figure 4.12 (a) Represent sign for word “EIGHT” .....	
Figure 4.12 (b) Represents HamNoSys for word “EIGHT” after selecting	44
Hand orientation.....	44
Figure 4.13 Hand Locations for generating HamNoSys.....	45
Figure 4.14 Details of head locations.....	46
Figure 4.15 (a) Represent sign for word “SAY” .....	46
Figure 4.15 (b) Represents HamNoSys for word “SAY”.....	46
Figure 4.16 Details of Mouth locations.....	47
Figure 4.17 (a) Represent sign for word “TEETH”.....	47
Figure 4.17 (b) Represents HamNoSys for word “TEETH”.....	47
Figure 4.18 Details of Paired locations.....	48
Figure 4.19 (a) Represent sign for word “SLEEP”.....	48
Figure 4.19 (b) Represents HamNoSys for word “SLEEP”.....	48
Figure 4.20 Details of Trunk locations.....	49
Figure 4.21 (a) Represent sign for word “EIGHT”.....	

Figure 4.21 (b) Represents HamNoSys for word “EIGHT” after selecting	49
Hand location.....	49
Figure 4.22 Hand position with respect to selected location.....	50
Figure 4.23 (a) Represent sign for word “EIGHT”.....	
Figure 4.23 (b) Represents HamNoSys for word “EIGHT” after selecting	50
Hand contact.....	50
Figure 4.24 Hand movements for generating HamNoSys.....	51
Figure 4.25 (a) Represent sign for word “EIGHT” .....	
Figure 4.25 (b) Represents HamNoSys for word “EIGHT” after selecting	51
Movements.....	52
Figure 4.26 Sign and HamNoSys for word “TWO”.....	53
Figure 4.27 Steps of HamNoSys generation for word “TWO”.....	

## List of Tables

---

Table 4.1 Categorization of number of symbols covered in HamNoSys	41
Generation system.....	52
Table 4.2 Steps to generate HamNoSys for word “TWO”.....	
Table 5.1 Example sentences used for testing of Punjabi sign language	55
Generation system.....	
Table 5.2 Words used for testing and their corresponding HamNoSys generated	59
By HamNoSys generation system.....	

# Chapter 1

## Introduction

---

### 1.1 Sign Language

Sign language (SL) is a visual gesture language which includes face, hands and arms to convey thoughts and meanings. It is developed in deaf communities which include deaf people, their friends and families. They use their hands, face, arm and body for communication. Sign languages are complete natural languages having their own syntax and grammar. Sign languages are not universal. These languages changes from one region to another. Every country is having their own sign language which varies from another country in syntax and grammar. Sign language which is used in America is known as American Sign Language (ASL) whereas sign language which is used in England is known as British Sign Language (BSL) and in India; Indian Sign Language (ISL) is used [1].

### 1.2 Indian Sign Language

Deaf people in India use ISL as their main mode of communication. It has been argued that same Indian Sign Language is used in Sri Lanka, Nepal, Bangladesh, and in some regions of Pakistan [2]. ISL have different dialects in different parts of India however they all are same grammatically. According to linguistic studies conducted in 1978, ISL is a complete natural sign language having its own phonology, morphology, syntax, and grammar [3]. The research on Indian Sign Language linguistics and phonological studies is limited because of lack of linguistically annotated and well documented data on Indian Sign Language. Indian Sign Language is used in deaf communities which include deaf people, hearing children of deaf adults, hearing parents of deaf children and hearing deaf educators. So, there is a need to build a system which can generate signs corresponding to the words of spoken language. It can help to learn Indian Sign Language. Hearing people can also use it to teach deaf students. Association of signs of different sign languages like ASL, BSL and ISL to a word will help the user to learn foreign sign languages simultaneously.

### 1.3 ISL Phonology

Indian Sign Language is a visual-spatial language which uses hands, face, arms, and head/body postures to provide linguistic information. To describe an event in Indian Sign Language, the signer uses 3D space around his body [4]. In case of spoken languages the communication medium is dependent on sound but in case of sign languages, the communication medium depends upon the visual channel. Unlike spoken language where a word is composed of phonemes, sign is composed of cheremes [5]. Cheremes are of two kinds manual and non-manual. A manual chereme includes hand shape, hand location, hand orientation and hand movements and non- manual chereme includes facial expressions, eye gaze and head or body posture [6]. There are some signs which contain only manual or non-manual components. For example, the sign of word “Yes” is performed with only vertical head node and there is no manual component for it [1].

Indian Sign Language signs can be classified into three classes: one handed signs, two handed signs, and non-manual signs. Figure 1.1 shows Indian sign hierarchy.

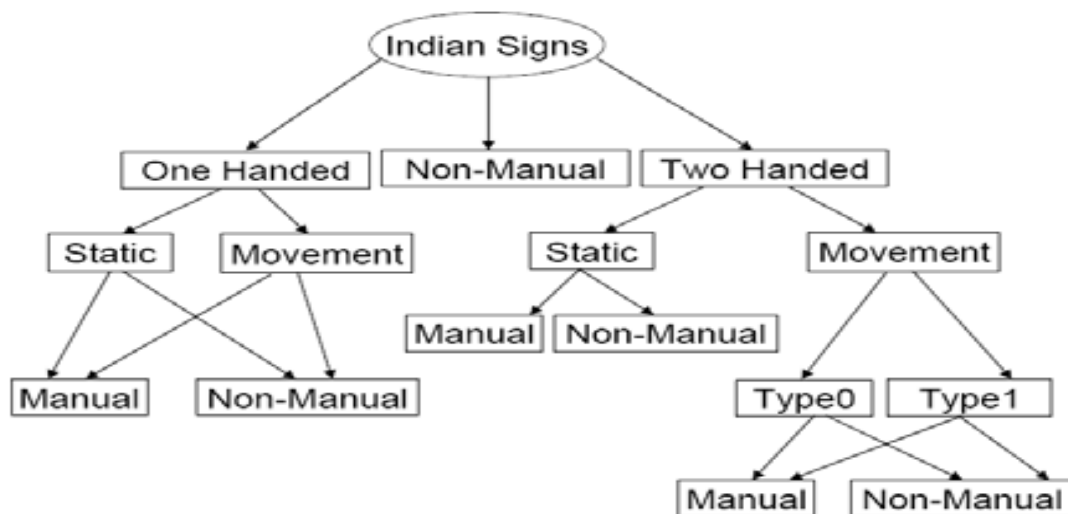
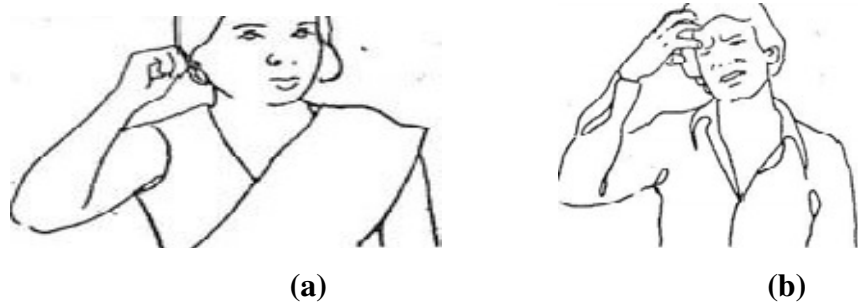


Figure 1.1: Types of Indian signs [1]

- **One handed signs**

The one handed signs are represented by using single dominating hand. These signs can be either static or movement related. The static and movement signs are further classified

into manual and non-manual signs. Figure 1.2 shows example of one handed static signs with both non-manual and manual components.

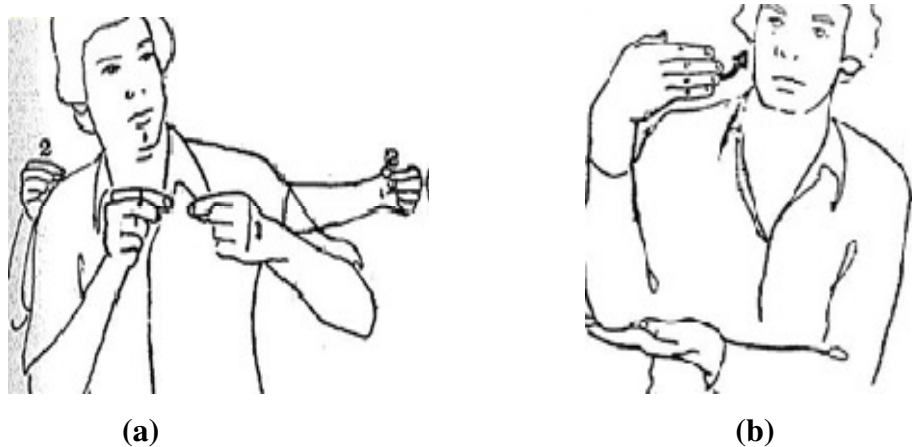


**Figure 1.2: (a) shows one handed static manual sign of Ear and (b) shows non-manual sign of headache [1]**

- **Two handed signs**

Classification for two handed signs is similar with classification of one handed signs. However, two handed signs with movements can be further distinguished as:

- I. Type0: Signs where both hands are active (as shown in figure 1.3(a)).
- II. Type1: Signs where one hand (dominant) is more active compared to the other hand (non-dominant) as shown in figure 1.3(b).



**Figure 1.3: (a) shows movement of both the hands for signing "long" and (b) shows the movement of only dominant right hand for signing "flag" [1]**

## 1.4 Challenges in Processing of Sign Language

Unlike spoken languages or written languages, sign language depends on the multiple simultaneous channels which include hand shape, hand location, palm orientation, hand or arm movement, facial expressions, and non-manual signals for communication [5]. For expressing supplementary meanings, sign language can modify different aspects of the manual signs which include hand shape, motion, path, timing and repetition for performing additional grammatical facial expressions, or it can use space around the signer. The space around sign language signer is used for several grammatical and descriptive purposes. Throughout the conversation, an entity which is under discussion can be arranged at a point in the signing space. For pointing to this location, reference to this entity will be made, and some of the verb signs will move toward this entity position or away from this to indicate their arguments. But choosing the location for make use of signing space is not meaningful, because during signing, one imaginary entity which is signed at left of another in the signing space does not assure that in real world this entity will actually be at the left of another entity.

For signing, translation between two languages is very difficult. It does not matter whether languages are related to each other or not. There are some cases where closely following source text gives unnatural or even unacceptable translation. This is the case with the translation between English and Indian Sign Language [7]. They differ typologically and in the medium of expression. The main difference is the role of pragmatics in the two languages. Indian Sign Language utterances reflect the immediate context in much more defines way than English language. Because of this, it is very difficult to provide Indian Sign Language translation for any given dialogue. This problem has been illustrated using example sentences given in 1.1 and 1.2.

Person A: Which Guide would you prefers? ...1.1

Person B: I don't mind. ...1.2

In Indian Sign Language, A will be dependent on the choices he will be having: if four people will be there, then firstly those four people will be identified. Then four different placements will be shown using the neutral space. After that, sign for word <WHICH?>

will be performed across the neutral space. Then for the response, the signer B would just point at each placement then sign <EITHER>, then <DON'T MIND>. Signing only for <DON'T MIND> without any context would be misleading and meaningless.

Open ended questions in English are better translated into Indian Sign Language with a range of possible answers [7]. For example, example sentence given in 1.3 has been translated into example sentence given in 1.4.

Person A: How long will it take you to get here? ...1.3

Person B: YOU-GET-HERE WHAT TIME? 15 MINUTES? 10 MINUTES? ...1.4

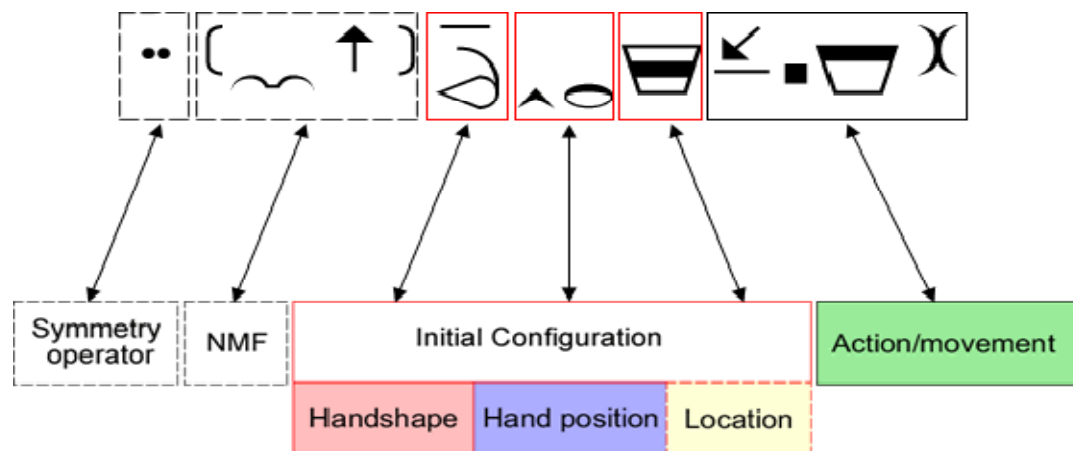
### **1.5 HamNoSys: A Notation System for Sign Language**

Sign language does not have a particular written form. For defining a sign there must be a notation system that can help to write signs. The Hamburg sign language notation system known as HamNoSys is one such system [8]. It uses phonetic transcription system to transcribe signing gestures. HamNoSys is a syntactic representation of a sign which provides computer processing for signs [9]. HamNoSys has its roots in the Stokoe notation system which introduced an alphabetic system to describe the sub lexical parameters including hand location, hand configuration and hand movement to give a phonological description of signs [10]. First version of HamNoSys was defined in 1984. Objectives of HamNoSys are given as below.

- a) HamNoSys transcriptions should be useful for all sign languages in the world, and the notations should not rely on conventions which differ from country to country.
- b) Meaning of alphabets should be clear and easy to remember for the users.
- c) It should be possible to transcribe any signed utterance with HamNoSys; notation of majority of signs should make use of such principles which results in shorter notation for the average sign.
- d) The notation system should be usable in standard text processing and database applications with their computer supported transcription.
- e) HamNoSys should allow both for a general evolution and specializations. New versions of the system should be compatible with the present ones.

## 1.6 Importance of HamNoSys in Generation of Sign Language

HamNoSys is a Stokoe based notation system which is having 200 largely iconic characters [11]. These HamNoSys symbols are used to represent hand orientation, hand shape, and hand location corresponding to other parts of the body. For writing signs, these parameters should be assigned with some value. HamNoSys provides symmetry operator for representing two handed signs non-manual components. Non-manual phonological can be represented by replacing hand graphemes with the symbols for body part such as head [12]. But facial movements like raised eyebrows or puffed out cheeks are complex to represent. The parameters of a sign are written in the order of symmetry operator, non-manual components, hand shape, hand position, hand location and hand movement as shown in figure 1.4.



**Figure 1.4: Structure of HamNoSys [12]**

HamNoSys for a single sign describes the initial posture of the non-manual features, hand shape, hand orientation and hand location plus the actions which change this posture in a sequence or parallel. In case of two handed signs, initial posture notation is preceded by the symmetry operator that defines how the description of the non dominant hand gets copied into the dominant hand.

### Example

Traditional Stokoe based notation system includes only basic hand parameters. HamNoSys has extended it by expanding sign representation parameters. These parameters include:

- a) Dominant hand's shape,
- b) Location of the dominant and the non-dominant hand with respect to the body,
- c) Extended finger orientation of both dominant and non-dominant hand,
- d) Palm orientation of both hands,
- e) Movements (straight, curved, or circular) and,
- f) Non-manual signs.

Figure 1.5 shows examples of different HamNoSys symbols and their descriptions.

Symbol	Description
	index finger stretched
	extended finger ahead
	palm orientated left
	location shoulder height
	fully stretched out
	hand move ahead
	hand move right

**Figure 1.5: HamNoSys symbols and there descriptions [1]**

Figure 1.6, shows the HamNoSys of the word “WOMAN” whereas its sign is shown in figure 1.7.

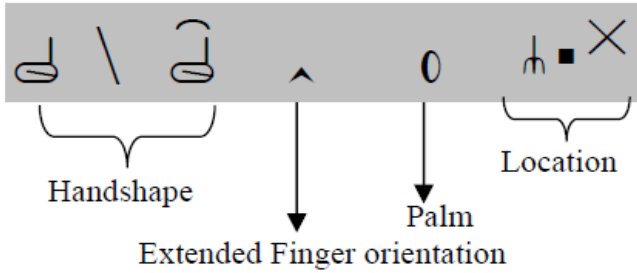


Figure 1.6: HamNoSys of “WOMAN” [1]



Figure 1.7. Sign of “WOMAN”[1]

HamNoSys creates language independence because it is not specific to a particular Sign Language, it can be used to describe any signs of any language. HamNoSys creates visual phonetics of signs which are language independent [13]. HamNoSys is used as the basis for Signing Gesture Markup Language (SiGML), which is further used to generate animation using animated signing avatar.

### 1.7 SiGML Language

SiGML is Signing Gesture Mark-up Language. It was developed for specifying signing sequences in ViSiCAST project. It defines HamNoSys symbols into XML tags form. It was developed at the University of East Anglia. It provides communication tools in form of animated figures. SiGML representation made from HamNoSys notation of sign language is readable by 3D rendering software [14]. SiGML for word “DEAF” is shown in figure 1.8.

```

<sigml>
  <hns_sign gloss="Deaf">
    <hamnosys_nonmanual>
    </hamnosys_nonmanual>
    <hamnosys_manual>
    <hamfinger23/>
    <hamextfingerui/>
    <hambetween/>
    <hamextfingerul/>
    <hampalmd/>
  </hns_sign>
</sigml>

```

```
</hamear/>
</hamtouch/>
</hamnosys_manual>
</hns_sign>
</sigml>
```

**Figure 1.8: SiGML for word “DEAF”**

## 1.8 Avatars

Avatars are described as “digitally created humanoids” or “virtual bodies”. They take input as SiGML language or XML text and generate animation corresponding to inputted text. Animation frame definitions are given as input to avatar in a sequence. They describe the static pose of the avatar. These sequences also describe the avatars time stamp, *i.e.*, at what time avatar will be placed in that pose [15]. When these avatars are placed in sequence of poses then rendering software produces signing animation corresponding to specified frame definitions. Many avatars have been developed for generating sign animations. They are given names as “Anna”, “Marc” and “Francoise” as shown in figure 1.9.



**Figure 1.9: Different Avatars with their names**

## **1.9 Thesis Outline**

This thesis is divided into 6 chapters. Chapter 1 includes introduction of sign language and Indian Sign Language. It describes ISL phonology, HamNoSys, SiGML and avatar. Chapter 2 gives details of the systems that have been developed on sign language generation. It includes machine translation systems, Dictionary tools and automated sign language generation systems. Chapter 3 includes the problem statement, objectives and methodology for developing system on sign language. In chapter 4, sign language generation system for Punjabi language is proposed. This chapter also includes the details of the implementation of HamNoSys generation system. Results of these systems are discussed in chapter 5 and chapter 6 concludes the work done in this thesis.

### **Chapter Summary**

In this chapter, sign language is introduced along with Indian Sign Language. ISL phonology is described with its types and signs. For developing a system for sign language, which challenges can arise is covered. Notation system for sign language means HamNoSys is also introduced with an example. Brief description about SiGML and Avatar is given.

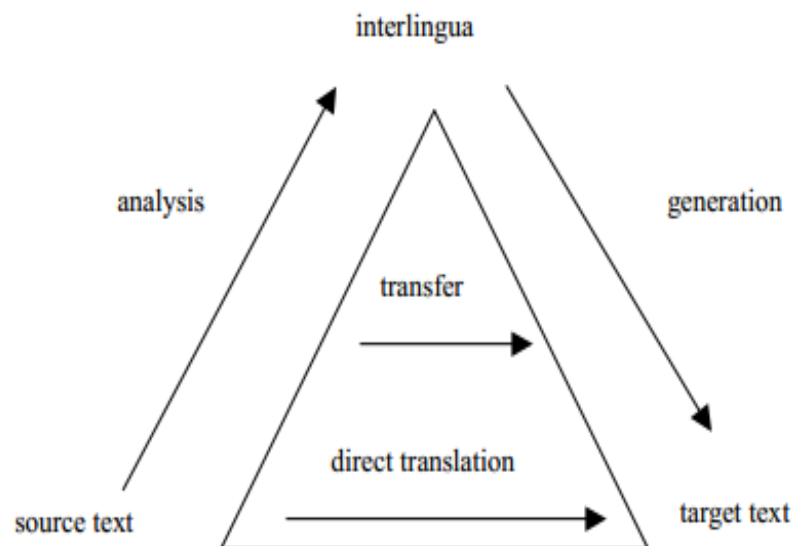
## Chapter 2

### Review of Literature

---

Sign languages are as diverse as spoken languages. Deaf people in different countries use different sign language for communication, but some sign languages have a similar structure. Each language have their own signs which becomes different from signs used in any other region's language, so for the communication with different region's people its beneficial to have a automated system which would be able to translate our spoken language into sign language. These systems are known as machine translation (MT) systems [16]. These systems take Punjabi or English language as input, process it and translate it into target sign language. There is a spectrum of architectural designs along which most machine translation systems can be classified, and loosely they are classified into three basic designs as shown in figure 2.1 [17].

- Direct Translation System
- Transfer Based Architecture
- Interlingua



**Figure 2.1: Traditional Machine Translation pyramid [18]**

## **2.1 Direct Translation Systems**

These systems perform their processing on the individual words of the source language string; translation is achieved without performing any form of syntactic analysis on the original input text. Generally the word order of the sign language remains the same as that of the English text. But in the case of English to Indian Sign Language, target sign language may not allow the same word order. With this, system also assumes a strong knowledge of both the English as well as the target sign language. Systems based on this approach are TESSA [19] and SignSynth project [20].

## **2.2 Transfer Systems**

These systems analyze the input text to some syntactic or semantic level, according to that a special set of transfer rules are employed which read the information of the source language structure and produce a syntactic or semantic structure in the target language. Afterwards, a generation component is used to convert this linguistic structure into a target language surface form. The transfer grammar approach is used in text to SL MT systems and also in text to text MT systems. Systems based on this approach are TEAM [21], ASL workbench [18] and ViSiCAST translator [22].

## **2.3 Interlingual Systems**

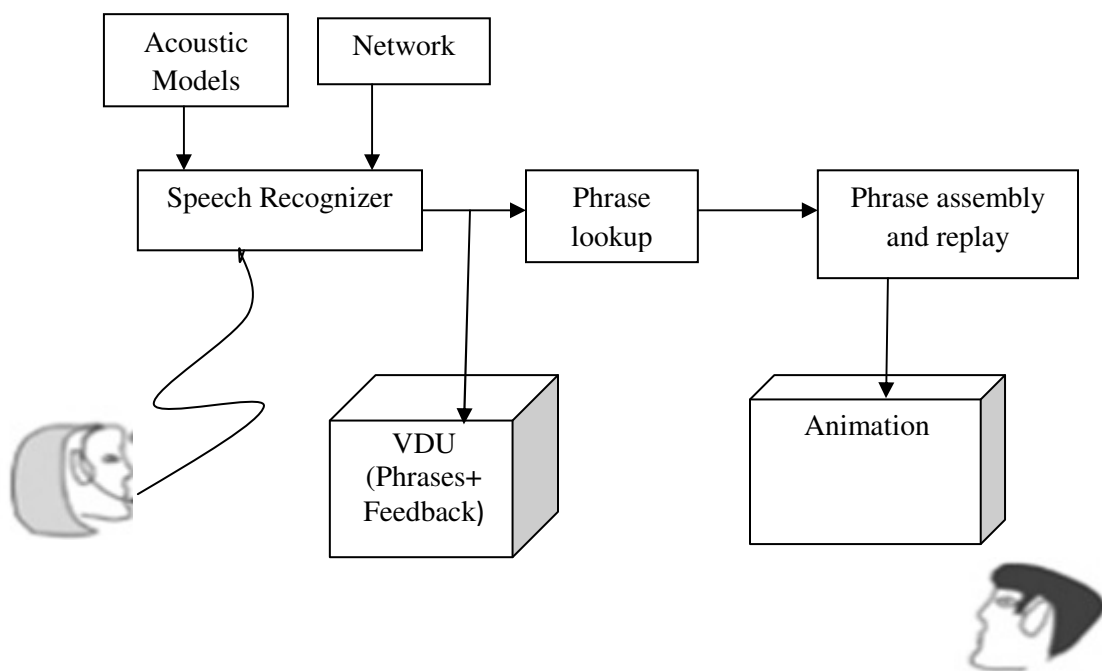
In these systems, the source is analyzed and processed semantically to produce a typically language independent semantic representation structure called an Interlingua, and then a Generation component produces the target language surface form from this starting point. A system based on this approach is ZARDOZ system [23].

## **2.4 Existing Systems on Sign Language**

Researchers all over the world have been working on automatic generation of sign language. These automated systems take text or speech as input and produce animation for it.

### 2.4.1 TESSA

Cox *et al.* (2002) had developed TESSA system based on direct translation approach. It is a speech to British Sign Language translation system which provides communication between a deaf person and a post office clerk. TESSA takes English as input text, lookup each word of the English string in the English-to-Sign dictionary, concatenates those signs together, and blends them into an animation. In this system, formulaic grammar approach is used in which a set of predefined phrases are stored for translation and translated by using a phrase lookup table. Architecture of system is shown in figure 2.2.

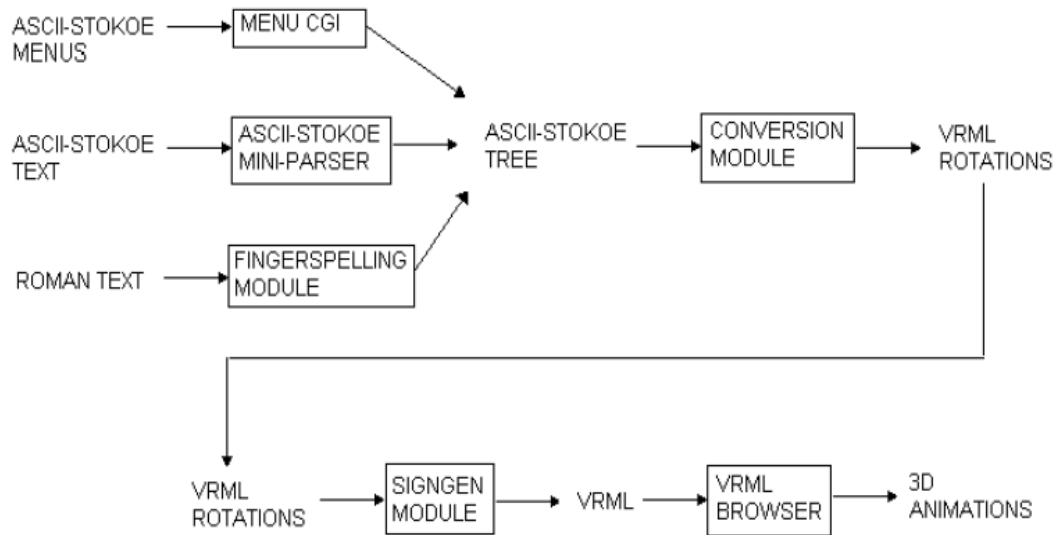


**Figure 2.2: Structure of TESSA system [19]**

The post office clerk uses headset microphone and speech recognizer. In speech recognizer, legal phrases from the grammar are stored. When clerk speaks a phrase, speech recognizer matches it with legal stored phrases. Clerk's screen displays topics available corresponding to uttered phrase, for example, "Postage", "Bill Payments" and "Passports". From these phrases clerk select one phrase according to requirement and sign of that phrase is displayed on the screen. Because of use of a small set of sentences as templates TESSA is a very domain specific system. Currently there are around 370 phrases stored in this system [19].

### 2.4.2 SignSynth Project

It is a prototype sign synthesis application which is under development at the University of New Mexico. It converts input text into American Sign Language. Sign synthesis and speech synthesis performs almost same task. The only difference is in the form of outputs. Thus the architectures of both of these are also almost same. Sign synthesis uses Perl scripts through the common gateway interface (CGI) for performing signing animation. Architecture of SignSynth is shown in figure 2.3.



**Figure 2.3: Architecture of SignSynth system [20]**

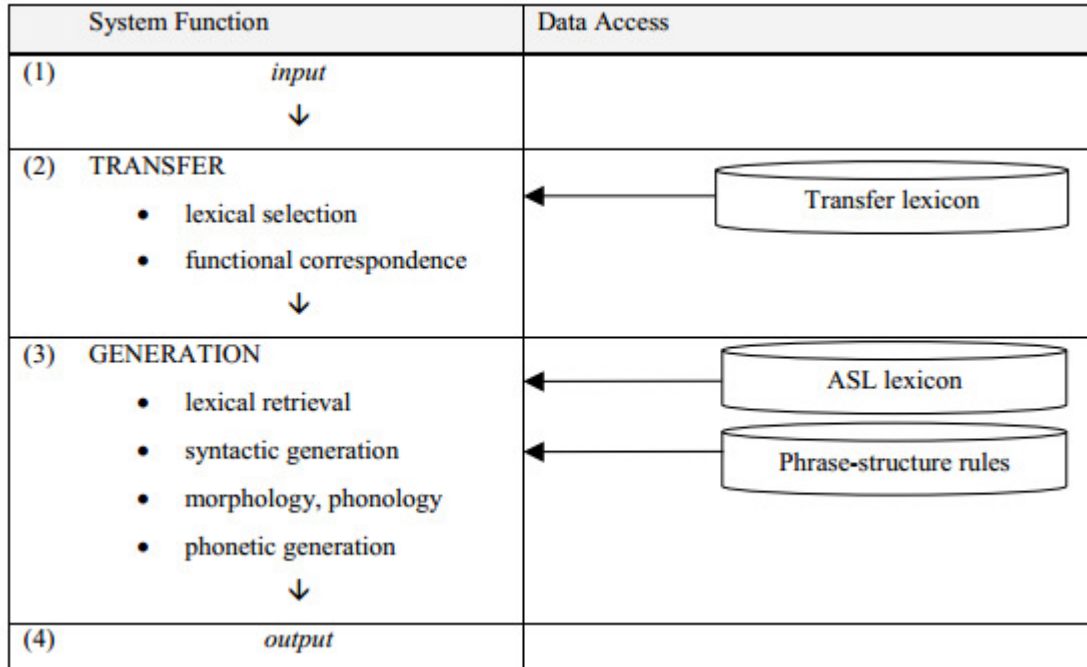
It has three main interfaces. First interface, MENU CGI offers menus for signs by which user can specify the phonological parameters. Additional menus help such users who know nothing about ASCII-Stokoe. They can directly select the hand shape, hand location and hand orientation for each hold. Second interface, ASCII-Stokoe mini-parser is for more advanced users to type with additions for timing and non-manuals. The finger spelling module helps the user to type in the Roman alphabet. This module outputs an ASCII-Stokoe tree which becomes as input to the conversion module. Conversion module further produces Web3D rotations for joints which are used. After the creation of rotations, they become the input for SignGen module. SignGen module integrates them with Web3D humanoid for creating complete web file with animation data. Then with the help of plug-in, animation is played. SignSynth is free and open-source. It has simple humanoid and Perl CGI which runs on any web server [20].

### **2.4.3 TEAM**

Zhao *et al.* (2002) had developed a system TEAM based on machine translation. It is a translation system which converts English text into American Sign Language. In this system TAG parser analyzes the input text. It involves two major steps. First step includes the translation of input English sentence into intermediate representation. It considers syntactic, grammatical and morphological information. In second step, its interpretation is performed. Its representation is done as motion representation which actually controls the human model and produce ASL signs. This system uses gloss notations for generating intermediate representation. Firstly it analyzes the word order and then generates glosses which represent facial expressions, sentence types, and morphological information. It uses a synchronous tree adjoining grammar (STAG) to map information from English text to ASL. It is first translation system which considers visual and spatial information along with linguistic information associated with American Sign Language. It is not limited to ASL; it is expandable to other signed languages because of its flexibility [21].

### **2.4.4 The ASL Workbench**

Speers (2001) had proposed and implemented a system ASL workbench. It is a Text to ASL Machine Translation system. It analyses the input text up to an LFG-style f-structure. Its representation abstracts some of syntactic specifics from input text and replaces them with linguistic features of the text. Architecture of ASL workbench is shown in figure 2.4. Workbench system included a set of specially written rules for translating an f-structure representation of English into one for ASL. It implements transfer module and generation module. Input for translation module is an English LFG f-structure. It is converted into an ASL f-structure using structural correspondence and performing lexical selection. The ASL f-structure becomes input to the generation module. Generation module creates American Sign Language c-structure and p-structure corresponding to the sentence. ASL workbench performs fingerspell of the word if the lexical element is noun.

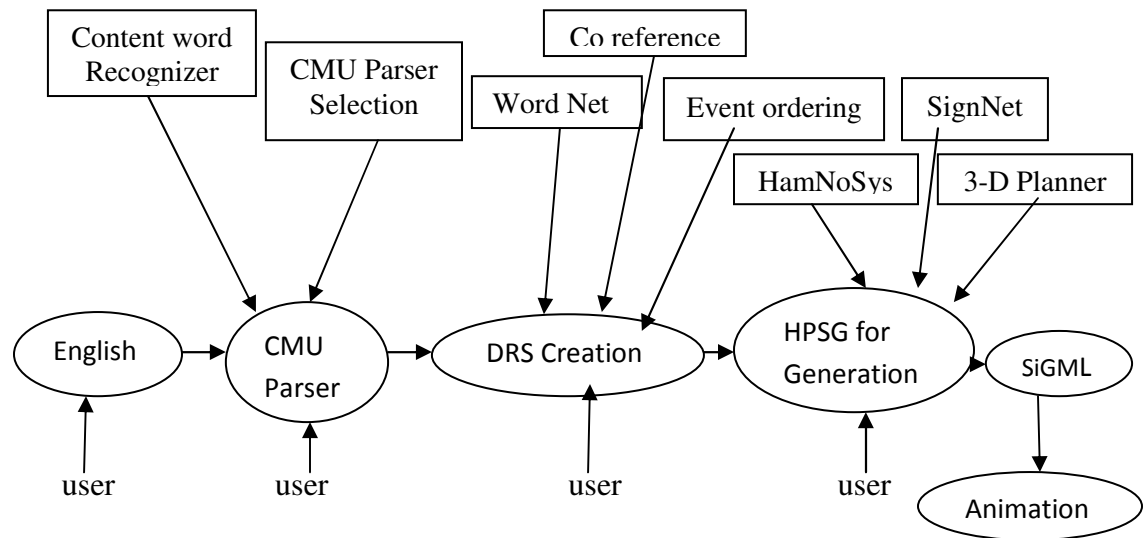


**Figure 2.4: Architecture of ASL workbench [18]**

If element is other than noun then translation fails, although translator can create entry corresponding to the word in ASL lexicon. It can also create entry in transfer lexicon if necessary and re attempt the translation [18].

#### 2.4.5 ViSiCAST Translator

Safar and Marshall (2001) had developed English to British Sign Language translation system. It uses semantic level of representation for performing English analysis to the BSL generation [22]. It includes investigation of sign language delivery using different technologies. The architecture of ViSiCAST is shown in figure 2.5. This system is user friendly. User enters English text into the system, at this stage user can change or alter the text according to the requirement. After that, at the syntactic stage inputted text is parsed with CMU (Carnegie Mellon University) link grammar parser [25]. From this parser, an intermediate semantic representation is made in the form of Discourse Representation Structure (DRS) [26]. From this representation, morphology and syntax of sign generation is defined in Head Driven Phrase Structure Grammar (HPSG) [27]. Here, signs are shown in form of HamNoSys and can be edited.



**Figure 2.5: Architecture of ViSiCAST system [22]**

After this, signing symbols are linked with SiGML, which describes signing notations into XML form. SiGML is easily understandable by 3D rendering software, which plays animation corresponding to inputted text.

#### 2.4.6 ZARDOZ

Veale *et al.* (1998) had developed a system ZARDOZ based on Interlingua approach. It is English to Sign Languages translation system which uses a set of hand coded schemata as an Interlingua for a translation component. Architecture of ZARDOZ system is shown in figure 2.6. It illustrates the blackboard compartmentalization of ZARDOZ system into panels. Working of these panels is given below.

Step I: Input text is processed by lexical expert demons. They apply morphological rules on text to recognize compound word constructs.

Step II: Here, idiomatic reduction is applied on text.

Step III: Text is parsed to produce deep syntactic/semantic representation.

Step IV: Interlingua representation is composed from unification structure.

Step V: Schematization removes metaphoric and metonymic structures which are specific to the source language.

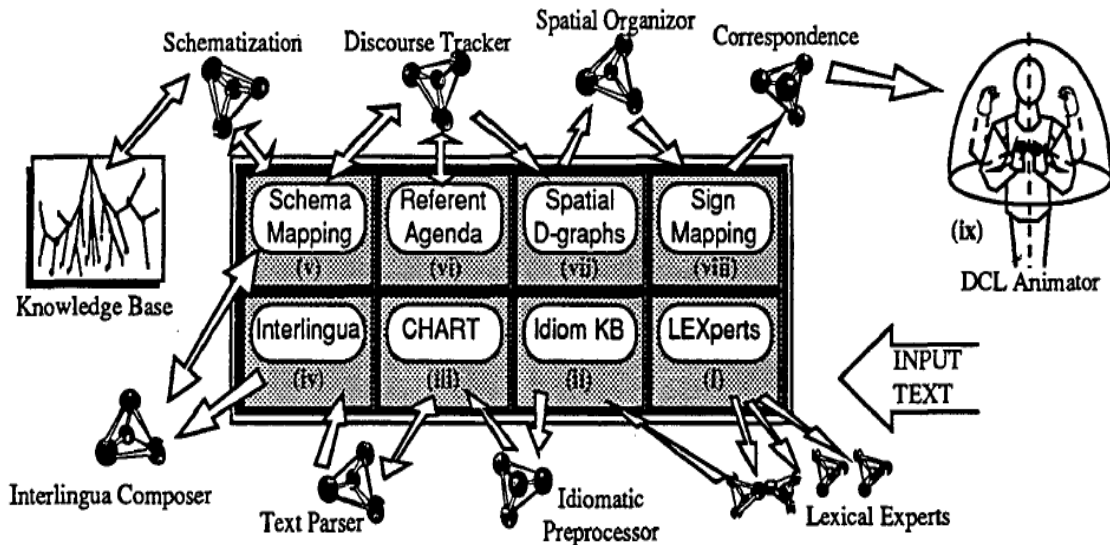


Figure 2.6: Architecture of ZARDOZ system [23]

Step VI: The refined Interlingua provides grist for discourse tracking agency.

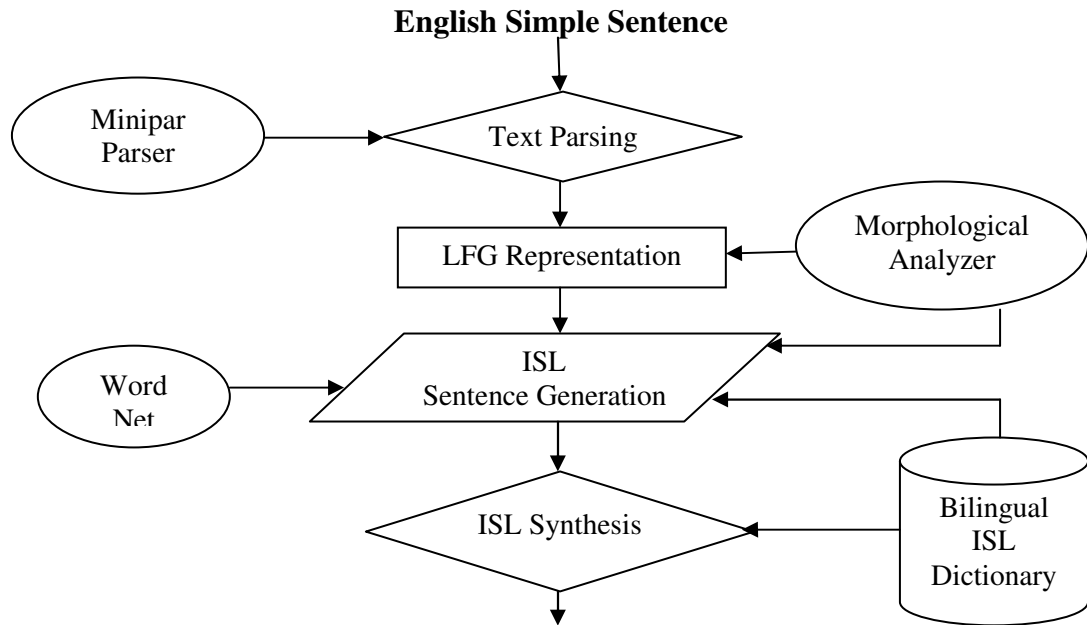
Step VII: It performs anaphoric resolution. Then spatial dependency graphs are generated by sign syntax agency.

Step VIII: Sign mapping agency assigns concept-to-sign corresponding to the tokens. Then syntax and mapping agencies convert Interlingua structure into sign tokens. Doll Control Language (DCL) animator compile these tokens into a DCL program. It controls on screen animation to correct gesture sequence which is visible to end user [23].

#### 2.4.7 Machine Translation System from Text-To-Indian Sign Language

Dasgupta *et al.* (2008) had developed a system based on machine translation approach. It takes English text as input and generates signs corresponding to the inputted text. Architecture of system is illustrated in figure 2.7.

Figure 2.7 shows four essential modules of the system which are, input text preprocessor and parser, LFG f-structure representation, Transfer Grammar Rules, ISL Sentence Generation and ISL synthesis. Simple English sentence is inputted to the parser. Simple sentence means which sentence has only one main verb in it. Minipar parser parses the sentence and make dependency tree. A phrase lookup table of around 350 phrases and temporal expressions is made before parsing. English morphological analyzer identifies plurality of nouns.



**Figure 2.7: Architecture of Text-To-Indian Sign Language machine translation system [17]**

LFG functional structure (f-structure) encodes grammatical relation of the input sentence. It also includes the higher syntactic and functional information representation of a sentence. This information is represented as a set of attribute-value pairs. Where attribute is for name of a grammatical symbol and value is for feature possessed by the constituent. It becomes as input to the generation module which apply transfer grammar rules on it so that it could transfer source sentence into target structure. Lexical selection and word order correspondence are two main operations that are performed during generation phase. Lexical selection is done using English to ISL bilingual lexicon. For example, word like “BREAKFAST” in English is replaced by “MORNING FOOD” in ISL. ISL uses Subject-Object-Verb (SOV) word order. Example sentence given in 2.1 shows the change in word order in ISL.

English “I have a LAPTOP” ...2.1  
 ISL “I LAPTOP HAVE”.

The final Indian Sign Language structure is achieved by the addition or deletion of words and restructuring of source representation [17].

## 2.5 Available Tools on Sign Language

Sign languages vary from one region to another in their syntax and grammar. Therefore, if different region's people want to communicate without learning signs then they need a dictionary that associates signs to the words of the spoken language of their region as well as dictionaries which will translate signs from one sign language to another. Some of these tools are given as:

### 2.5.1 A Multilingual Multimedia Indian Sign Language Dictionary Tool

Dasgupta *et al.* (2008) had developed a multilingual multimedia dictionary tool for Indian Sign Language. It is developed to create a GUI using which user can associate signs corresponding to the inputted text. Architecture of the system is shown in figure 2.8.

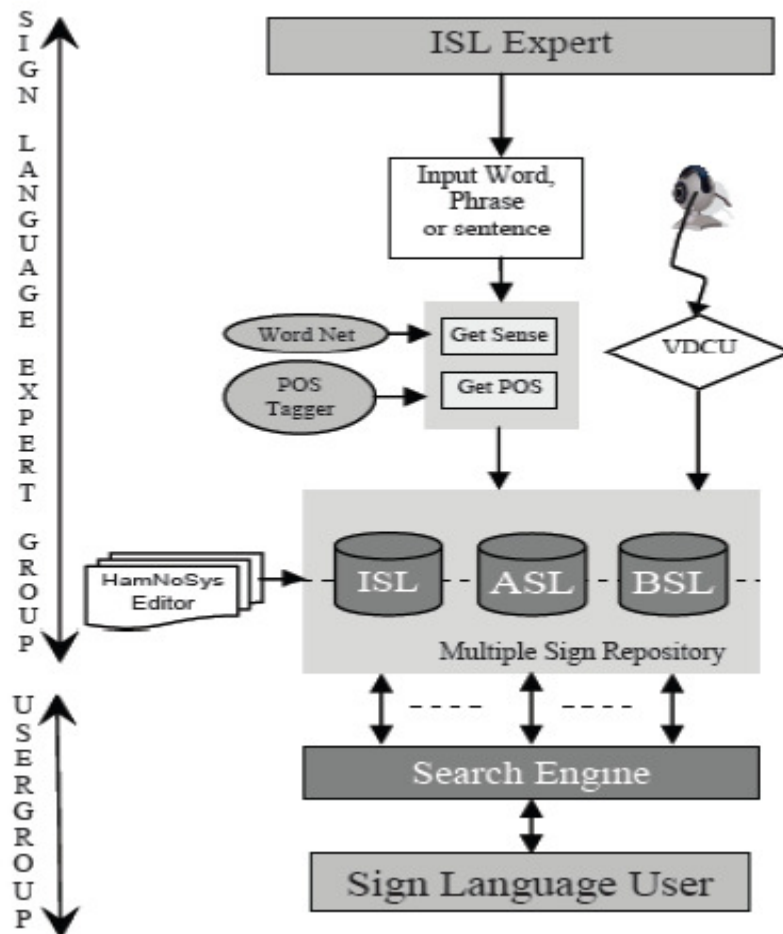


Figure 2.8: Architecture of Indian Sign Language dictionary tool [1]

As shown in figure 2.8, the system has two main modules, Expert module and User module. Expert module has further three units, *i.e.*, input text processing unit, visual data capture unit (VDCU) and sign storage unit [1]. In Input Text Processing Unit, Sign language expert chooses spoken language as input and target sign language. Then text is inputted into the system. Input text can be phrase, word or sentences. If inputted text is a word then system gives all possible meanings corresponding to the word with the help of WordNet and part of speech (POS). For having exact part of speech, sign language expert has to enter an example sentence along with the word. This sentence becomes an input to the POS-tagger. WordNet can return multiple senses for one word. User can select according to his requirement.

Visual data capture unit (VDCU) is connected through multiple webcams. Sign signed by signer is captured by VDCU. Webcams of VDCU are connected at different positions thus different articulation points of a sign can be stored in database. Storage unit is used to store Input text, digital video sign and phonological parameters corresponding to sign. These are stored in form of HamNoSys and further converted into XML formatted file [1].

### **2.5.2 Multimedia ASL Dictionary Tool**

Wilcox *et al.* (1994) had developed the multimedia dictionary. It is a Macintosh application which works as a bilingual (American Sign Language-English) dictionary for American Sign Language (MM-DASL). It presents American Sign Language signs in full motion digital video. Its major functions include the capability to search American Sign Language signs from entered English words and its capability to perform fuzzy searching. For each American Sign Language lexical entry, this dictionary contains grammatical information, definitions of sign, usage notes and successful English translations [28].

### **2.5.3 Computer Generated Models for the World Wide Web**

Geitz *et al.* (1996) had developed models of manual alphabet hand shapes. It is a teaching tool having collection of 3D computer graphic models which represents ASL manual alphabet hand shapes in different locations. These computer graphic models have been

recorded in the Virtual Reality Modeling Language (VRML) for display with World Wide Web browsers. These browsers include Mosaic or Netscape and some VRML browsers such as Worldview or Web Space [29].

#### 2.5.4 Indian Sign Languages using Flex Sensor Glove

Solanki (2013) had developed an electronic hand glove for communication of deaf and dumb people. It is portable and very easy to handle. Its flowchart is shown in figure 2.9. This system uses five flex sensors which are fitted with length of each finger and thumb. Using different position of fingers and thumb, Indian Sign Language's alphabets are made. Using these alphabets, word forming is made and their output is shown on the LCD. The sensor output is in analog form so this is converted in digital form using ADC. Then ADC's output goes in microcontroller where programming is done. Then the output is shown on the LCD [30].

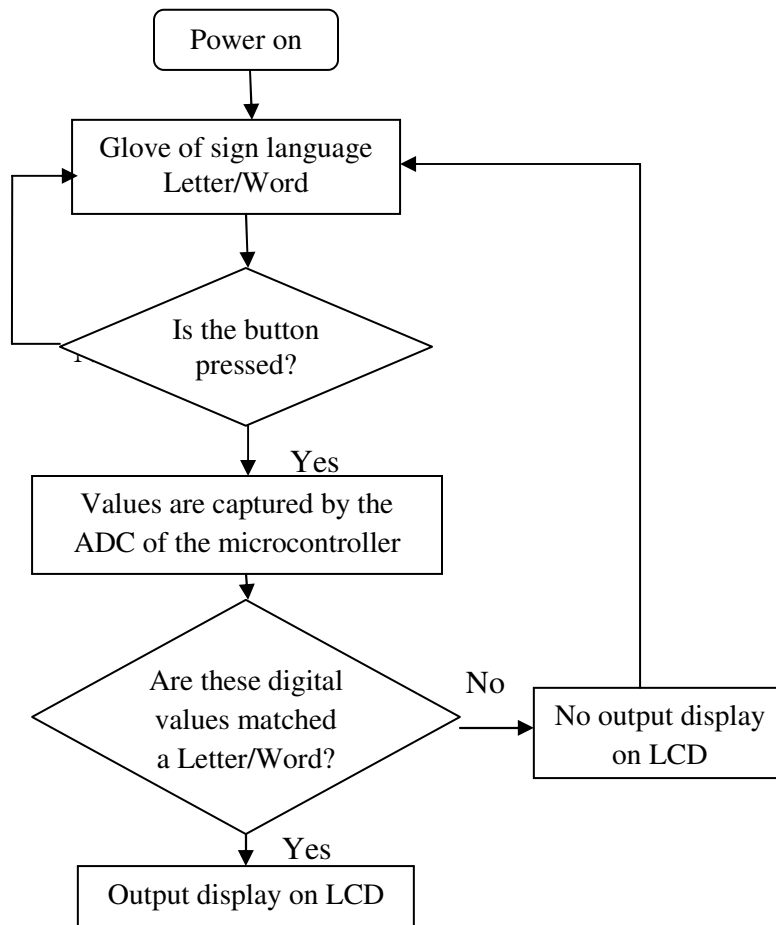
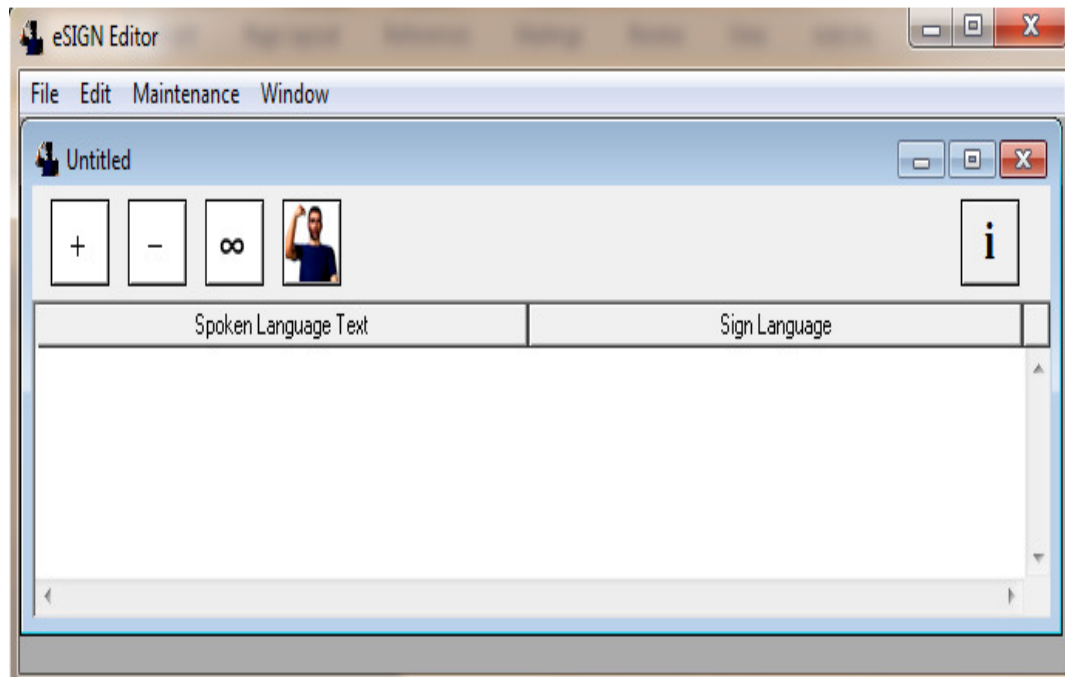


Figure 2.9: Flowchart of flex sensor hand glove [30]

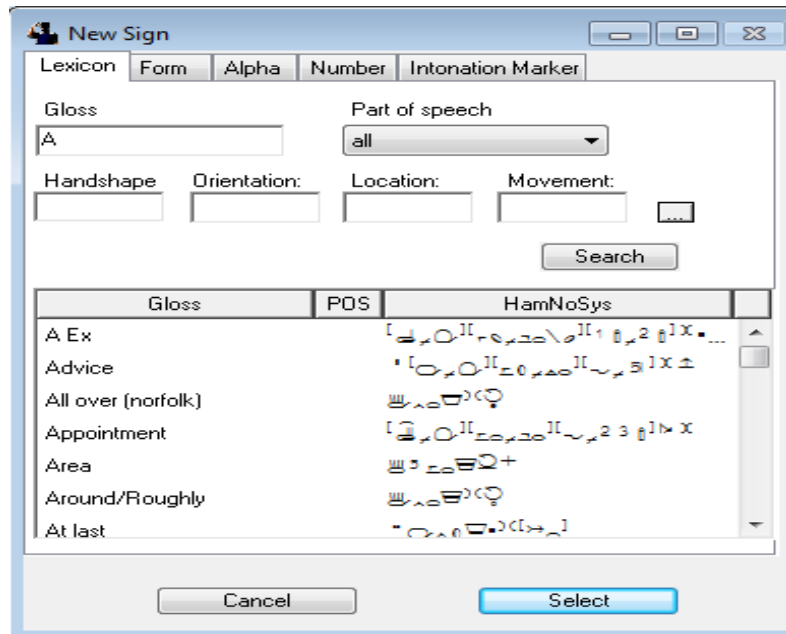
### 2.5.5 eSIGNEditor: An Editor of HamNoSys

An eSIGNEditor is an editor in which document represents signed text. In this, document window shows the signed text by sequences of glosses parallel to the spoken language text [31]. As shown in figure 2.10, the editor window is divided into two sections. In left column, phrases are entered and second column is for description of the sign translation. For adding any new description it have “+” button and for deleting description it have a “-” button as shown in figure 2.10.



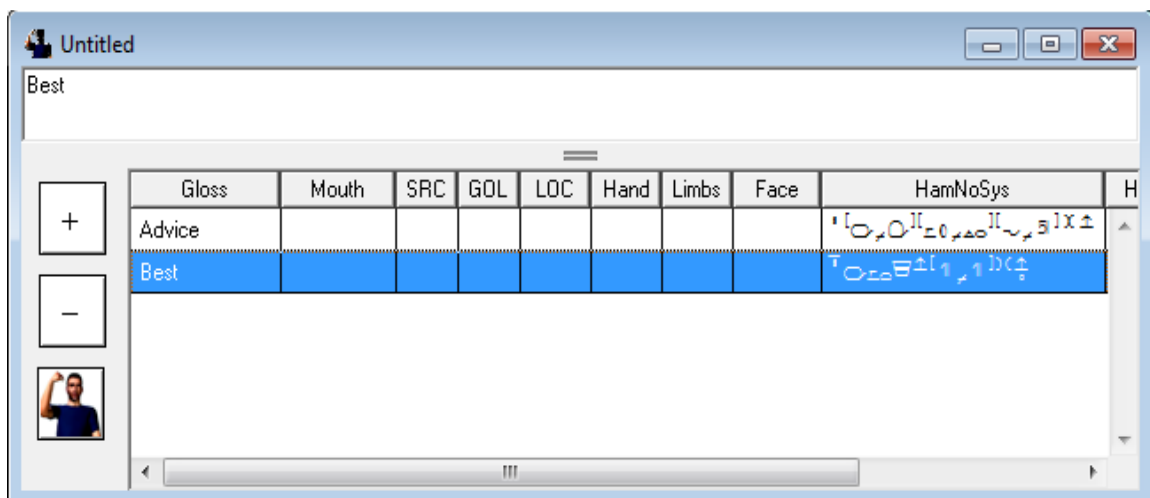
**Figure 2.10: eSIGNEditor interface**

Signs are written in form of HamNoSys in it. It stores generated signs into its database. Signs can be generated completely from the beginning, or by changing already generated signs. It provides search criteria for the signs which are stored in its vocabulary as shown in figure 2.11. Searching for words starting with letter “A” has shown in figure 2.11. For creating the signed contents, each utterance is handled separately.



**Figure 2.11: Searching word in eSIGNEditor**

In order to create or modify an utterance, the user opens a window to specify the sequence of signs as shown in figure 2.12. There is no requirement to change HamNoSys symbols manually. It uses the windows source (SRC, "Source"), destination (GOL, "Goal"), location (LOC, "Location"), and classification of the shape of the hand ("Hand") [32]. Changes can be made by double clicking on the corresponding cell as shown in figure 2.12.



**Figure 2.12: Document window of eSIGNEditor**

### 2.5.6 HamNoSys System for German Sign Language

HamNoSys is a well established transcription system developed by the institute for German Sign Language and deaf communication at the University of Hamburg for all SLs. HamNoSys is a phonetic notation system. It consists of about 200 symbols which covers the parameters of hand configuration, hand shape, hand location and hand movement. These symbols are iconic so one can easily recognize and learn them. The order of the symbols is fixed, but they still can transcribe a sign in lots of different ways. Because of phonemic notation the transcriptions are very precise. Notation for a single sign consists of a non-manual features, hand shape, hand orientation and location. For two handed signs, the initial posture notation is preceded by a symmetry operator that defines how the description of the dominant hand copies to the non dominant hand [32]. HamNoSys defines its symbols into three main categories. These categories are explained as:

- **Hand Shapes**

As shown in figure 2.13, hand shape describes symbols for basic hand forms and diacritics for thumb position and bending. Deviations from this description can be performed on individual fingers.

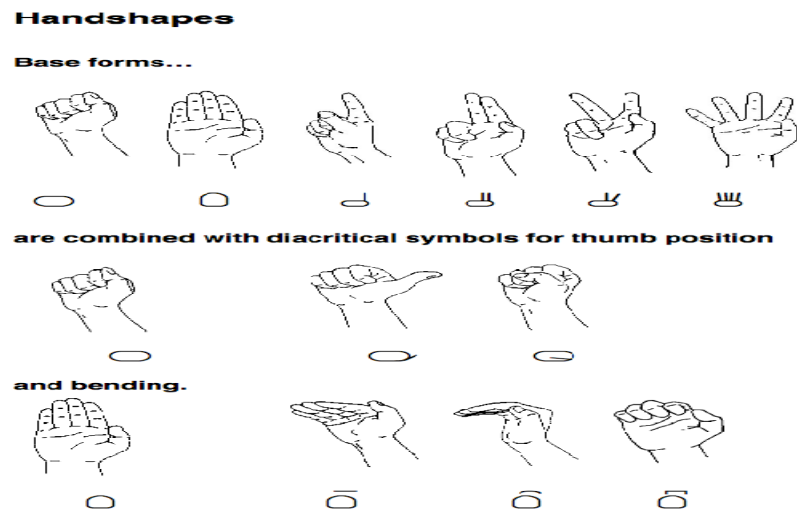
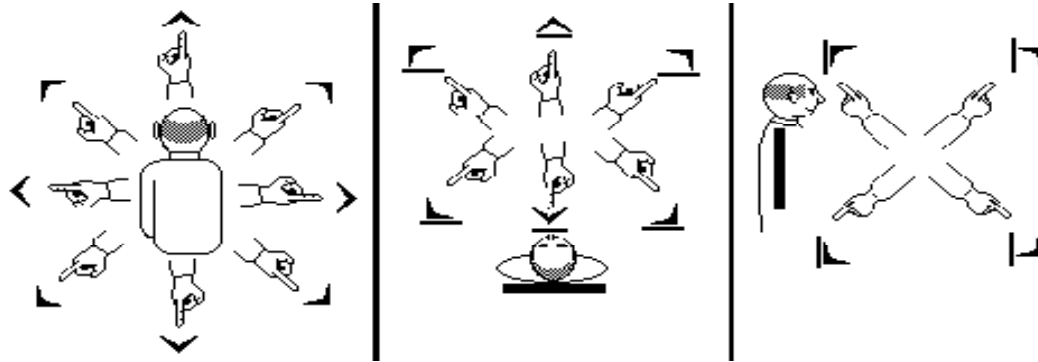


Figure 2.13: Hand shapes [32]

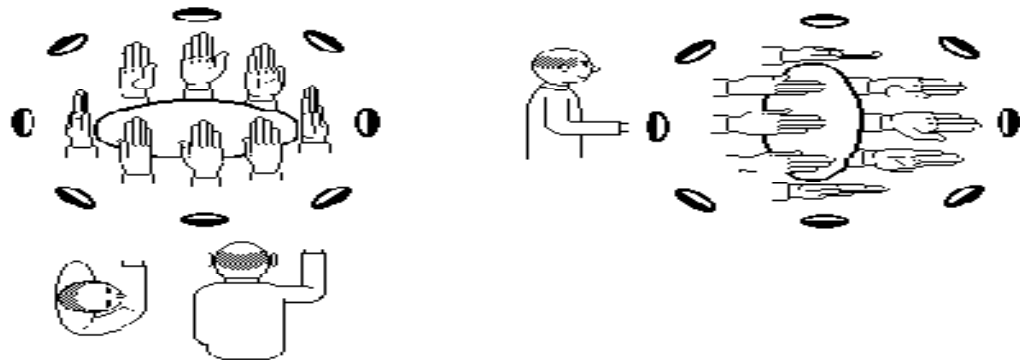
- **Hand Orientation**

Figure 2.14 shows the orientation of the hand by combining two components: extended finger direction specifying two degrees of freedom, and palm orientation describes the third degree.



**Figure 2.14: Hand orientation [32]**

For the third degree of freedom, only eight symbols are needed. The meaning of a symbol is defined relative to the extended finger direction, as shown in figure 2.15.



**Figure 2.15: Hand movements [32]**

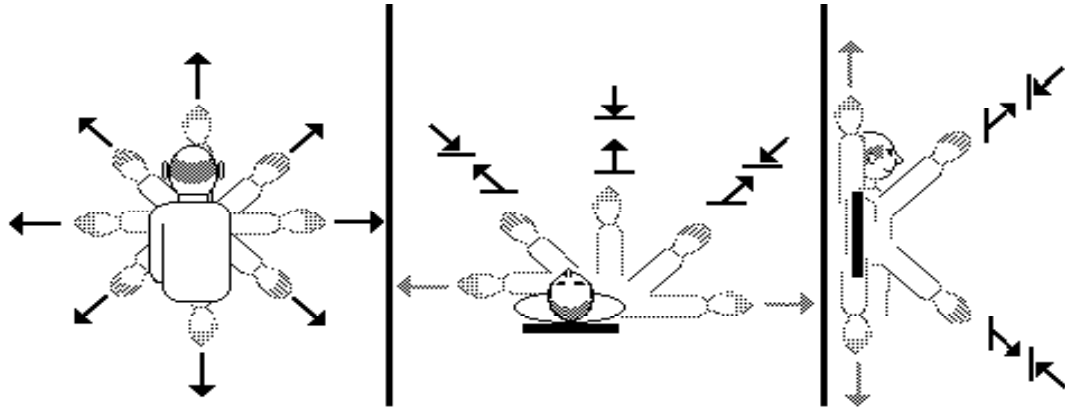
By adding a subscript, hand orientation can be made relative to the movement, for example, the palm orientation changes as the movement direction changes as shown in figure 2.16.



**Figure 2.16: Example of hand movement in "HOUSE" [32]**

- **Actions**

Actions are combinations of path movements and in place movements of the hands as well as nonmanual movements as shown in figure 2.17.



**Figure 2.17: Hand actions [32]**

Path movements can be specified either as targeted movements or relative movements. These are circles, straight lines, curved and zigzag lines, and similar forms. In place movements are changes in hand shape or hand orientation. A mode such as slow or sudden stop can be specified for each movement. Redundancy of actions can also be specified.

## **Chapter Summary**

This chapter includes explanation of sign language generation systems and dictionary tools on sign language. Systems which generate sign language from text are TESSA, SignSynth project, TEAM, ASL workbench, ViSiCAST translator, ZARDOZ, HamNoSys generation system and machine translation system from text to Indian Sign Language. Dictionary tools for sign language include multilingual multimedia ISL dictionary, Multimedia ASL dictionary tool, Flex sensor glove and eSIGNEditor.

## Chapter 3

### Problem Statement

---

According to All India Federation of the Deaf (AIFD), near about 4 million deaf and around 10 million hard of hearing people are there in India. And according to studies, one deaf out of every five deaf people in the world is from India. Indian Sign Language is a mode of communication for more than 1 million deaf adults and around 0.5 million deaf children [1]. However, according to UNESCO report (1980) there are only 5% of the deaf who are getting any education in India.

Many people who take birth in deaf families or born deaf, they learn sign language as their first language. Written form of sign language doesn't exist so writing or reading becomes less preferred language for such people. They prefer to access information in form of sign language on Internet rather than written content. But very few sites offer content in form of sign language. That content is available in form of video clips and can be accessed by reading the text. But, making available content on Internet using video approach is very expensive. It creates problems during downloading because of large files and it is also difficult to maintain it. If small change is required then there will be requirement to recreate it so it is not a promising approach.

So, providing web content in form of sign language, animation is best approach to use. And for developing animation written form of signs will be required. So, there is a need for system that can generate sign language from natural language text. Sign animation cannot be generated without HamNoSys. HamNoSys is based on Stokoe notation system. It represents signs in form of symbols. It has around 200 different symbols. It's not easy to remember all the symbols. For this, sign language expert is required having complete knowledge of HamNoSys. But in India, it is very difficult to find such an expert. Hence, there is a need for a GUI based HamNoSys generation system.

## **Objectives**

The main objective of this research work is to generate signs corresponding to text. In order to perform this task, following objectives were proposed to be carried out.

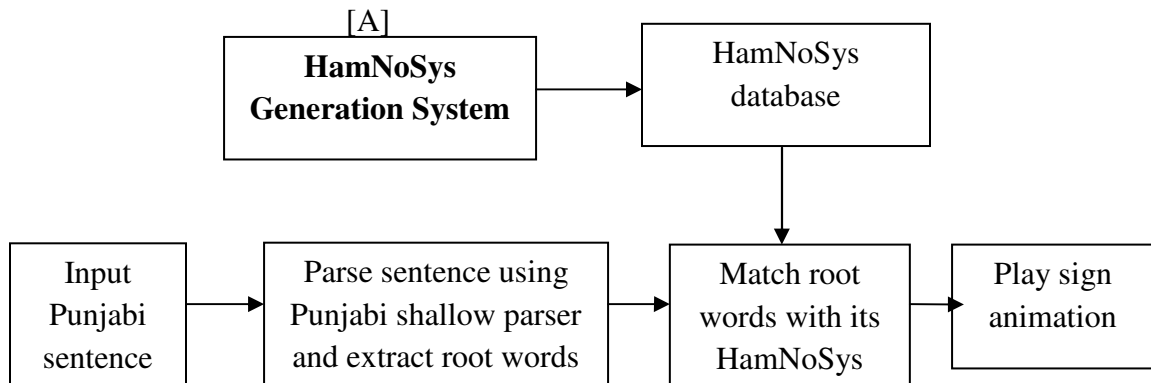
- Study of Indian sign language, HamNoSys and different available sign language tools.
- To understand the working of SiGML, Punjabi shallow parser, eSIGNEditor and URL App.
- To develop HamNoSys generation system.
- To develop sign language generation system for Punjabi language.

## **3.2 Methodology**

In this thesis, sign language generation system for Punjabi language is proposed which generates animation corresponding to text. It takes Punjabi sentence as input, extract root words from it and generate HamNoSys corresponding to root words. Then HamNoSys is converted into SiGML which generates animation for the sentence.

For generating HamNoSys, a system has been developed which generates HamNoSys corresponding to words. Anyone having little knowledge or no knowledge about HamNoSys can use this system to generate HamNoSys for different signs. This system firstly generates HamNoSys for signing words in form of hand shape, hand orientation, hand location and hand movements. Then it stores generated HamNoSys into its database.

Sign language generation system for Punjabi language uses HamNoSys generation system at [A] in figure 3.1. When system extracts root words from the sentence, then there is a need of HamNoSys corresponding to extracted words. HamNoSys generation system generates HamNoSys corresponding to small words and stores HamNoSys into database. From database, root words are mapped and required HamNoSys is selected. After that HamNoSys generated by HamNoSys generation system can be converted into SiGML and given as input to avatar module for producing sign animation.

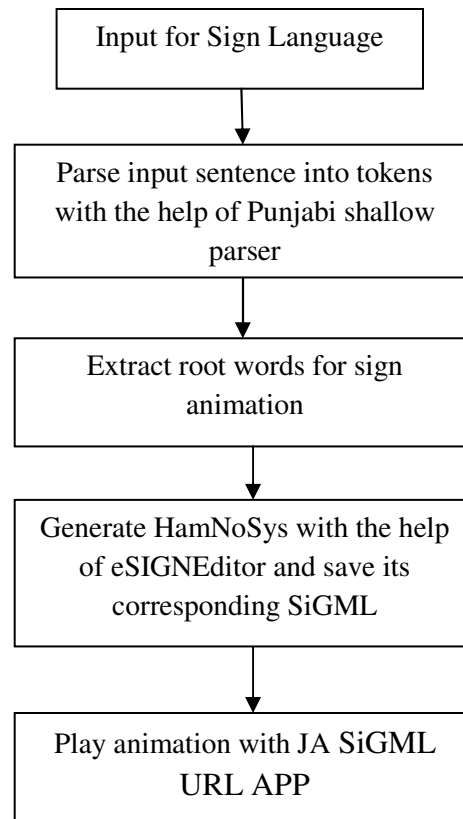


**Figure 3.1: Use of HamNoSys generation system in sign language generation system for Punjabi language**

Using these systems, online content can be transferred in form of sign language. Thus, it will be easy for deaf people to understand the content in form of signs rather than reading.

### **4.1 Architecture of Proposed Sign Language Generation System for Punjabi Language**

For generation of sign language from Punjabi input text, a system has been proposed in this thesis. In this system, Punjabi text is inputted by the user; system works on it and generates signs corresponding to inputted text. Architecture of proposed sign language generation system for Punjabi text has been shown in figure 4.1.



**Figure 4.1: Architecture of proposed sign language generation system for Punjabi language**

As shown in figure 4.1, the input Punjabi text is parsed using Punjabi shallow parser. Punjabi shallow parser has been developed by consortium of institutions headed by IIIT

Hyderabad, India. It is used for processing input Punjabi sentences [33]. It performs the tasks of tokenization, morph analysis, part of speech tagging and chunking for the processing of an input sentence. The working of Punjabi shallow parser has now been illustrated with an example sentence given in (4.1).

Punjabi Example sentence: ਛੋਟੇ ਬੱਚਿਆਂ ਨੇ ਕਿਤਾਬ ਪੜ੍ਹੀ। ... (4.1)

Transliterated example sentence: *chōṭē bacciām nē kitāb paṛhī.*

Equivalent English sentence: Little children read the book.

The output of Punjabi shallow parser for this example sentence at each intermediate stage has been explained in subsequent sub-sections.

#### 4.1.1 Tokenizer

A token is an instance of a sequence of characters in a sentence that are grouped together as a useful semantic unit for processing. The tokenizer converts a sentence into word level tokens consisting of words, punctuation marks, and other symbols. The output of tokenizer for the example sentence given in (4.1) is shown in (4.2).

Tokenizer:

-----

<Sentence id="1">

1	ਛੋਟੇ	unk
2	ਬੱਚਿਆਂ	unk
3	ਨੇ	unk
4	ਕਿਤਾਬ	unk
5	ਪੜ੍ਹੀ	unk

</Sentence>

... (4.2)

At this stage there is no part-of-speech information resolved for the tokens, so each token

has a ‘unk’ unknown token attribute.

#### 4.1.2 Morph Analyzer

The morphological analyzer identifies the root and the grammatical features of the word.

The output of morph analyzer for example sentence given in (4.1) is shown in (4.3).

Morph analyzer:

```
-----  
<Sentence id="1">  
1   ਛੋਟੇ   unk   <fs af='ਛੋਟਾ,adj,m,sg,,o,,'>|  
                                     <fs af='ਛੋਟਾ,adj,m,pl,,o,,'>  
2   ਬੱਚਿਆਂ   unk   <fs af='ਬੱਚਾ,n,m,pl,3,o,,'>  
3   ਨੇ       unk   <fs af='ਨੇ,psp,,,d,,'>|<fs af='ਆਂ,v,any,pl,2,,,>  
4   ਕਿਤਾਬ   unk   <fs af=' ਕਿਤਾਬ, n,f,sg,3,d,,'>  
5   ਪੜ੍ਹੀ     unk   <fs af=' ਪੜ੍ਹ,v,f,sg,any,,ਇਆ,ਇਆ>  
</Sentence>                                     ... (4.3)
```

Here, ‘fs’ is the feature structure which contains grammatical features of each word and ‘af’ is a composite attribute which consists of seven attributes, namely, root; lexical category; gender; number; person; case; tense, aspect and modality (TAM) information. In case, no value is given for a particular attribute the field is left blank. As shown in (4.3), ਛੋਟੇ *chōṭē* has a root word ਛੋਟਾ *chōṭā*; lexical category as ‘adj’ (adjective); gender as ‘m’ (male); number as ‘sg’ (singular) or ‘pl’ (plural), *i.e.*, correct number is not resolved at this stage, so, both singular and plural forms are shown in the feature structure; person is ‘not applicable’ for this token; case is ‘o’ (oblique) and TAM is ‘not applicable’ for this token.

### 4.1.3 Part-of-Speech Tagger

Part-of-speech tagging is the process of assigning a part-of-speech to each token in the sentence. It helps in analyzing the role of each constituent in a sentence. The output of part-of-speech tagger for example sentence given in (4.1) is shown in (4.4).

POS tagger:

```
-----  
<Sentence id="1">  
1    ਛੋਟੇ    JJ    <fs af='ਛੋਟਾ,adj,m,sg,,o,,'>|  
                                     <fs af='ਛੋਟਾ,adj,m,pl,,o,,'>  
2    ਬੱਚਿਆਂ NN    <fs af='ਬੱਚਾ, n, m, pl, 3, o,,'>  
3    ਨੇ      PSP   <fs af='ਨੇ,psp,,,,d,,'>|<fs af='ਆਂ,v,any,pl,2,,,,'>  
4    ਕਿਤਾਬ NN    <fs af=' ਕਿਤਾਬ, n, f, sg, 3, d,,'>  
5    ਪੜ੍ਹੀ    VM    <fs af=' ਪੜ੍ਹ,v,f,sg,any,,ਇਆ,ਇਆ'>  
</Sentence> ... (4.4)
```

As shown in (4.4), ਛੋਟੇ *chōṭē* 'little' is an adjective with symbol 'JJ', ਬੱਚਿਆਂ *bacciām* 'children' and ਕਿਤਾਬ *kitāb* 'book' are nouns with symbol 'NN', ਨੇ *nē* is postposition with symbol 'PSP' and ਪੜ੍ਹੀ *paṛhī* 'read' is the main verb with symbol 'VM' ( Bharati *et al.*, 2006).

### 4.1.4 Chunker

Chunking involves identifying noun phrases, verb groups, adjective phrase, and adverb phrase in a sentence. It involves identifying the boundary of chunks and the label. The output of chunker for example sentence given in (4.1) is shown in (4.5).

Chunker:

-----

<Sentence id="1">

1 (( NP

1.1 ਛੋਟੇ JJ <fs af='ਛੋਟਾ,adj,m,sg,,o,,'>|

<fs af='ਛੋਟਾ,adj,m,pl,,o,,'>

1.2 ਬੱਚਿਆਂ NN <fs af='ਬੱਚਾ, n, m, pl, 3, o,, '>

1.3 ਨੇ PSP <fs af='ਨੇ,psp,,,,d,,'>|<fs af='ਆਂ,v,any,pl,2,,, '>

)

2 (( NP

2.1 ਕਿਤਾਬ NN <fs af=' ਕਿਤਾਬ, n, f, sg, 3, d,, '>

)

3 (( VGF

3.1 ਪੜ੍ਹੀ VM <fs af=' ਪੜ੍ਹ, v, f, sg, any,, ਇਆ, ਇਆ '>

)

</Sentence>

... (4.5)

As shown in (4.5), the example sentence has three chunks, 'ਛੋਟੇ ਬੱਚਿਆਂ ਨੇ' '*chōṭē bacciām nē*' 'little children' and 'ਕਿਤਾਬ' *kitāb* 'book' as noun phrase 'NP' chunks and 'ਪੜ੍ਹੀ' *paṛhī* 'read' as finite verb chunk 'VGF'.

After performing tokenization, morph analysis, part-of-speech tagging and chunking, it produces final output by picking most appropriate morph with 'head' and 'vibhakti' computation. It uses the output of each intermediate stage and generates the output in 'Shakti' format. 'Shakti' format uses a common representation for the operation of all modules [34, 35]. From output file produced by Punjabi shallow parser, root words are extracted using java program. After extraction of words there is a need to write signs for these words into computer understandable language which is HamNoSys. eSIGNEditor is

a tool for performing this task. The main idea of this tool is to pick signed sentences sign by sign from the lexicon and apply morphological changes to individual signs or strings of signs which are necessary. An eSIGN Editor document represents a signed text. After writing signs HamNoSys code is converted into Signing Gesture Mark-up Language (SiGML). SiGML is a form of XML which defines a set of XML tags for each phonetic symbol in HamNoSys. SiGML files are described as plain text thus computer can easily handle them. Generated SiGML file is browsed by tool “JA SiGML URL APP”, which play sign animation for the inputted text.

## 4.2 Description of Proposed System with an Example Sentence

The process of generation of sign language from input Punjabi sentence has been illustrated with an example Punjabi sentence given in (4.1). The description of the proposed system has been explained by following different steps of processing of input text.

### Step 1: Create input file and parse it using Punjabi shallow parser

```

<Sentence id="1">
1 (( NP <fs
af='ਬੱਚਾ,n,m,pl,3,o,0_ਨੇ,' head='ਬੱਚਿਆਂ' vpos="vib2_3">
1.1 ਛੋਟੇ JJ <fs af='ਛੋਟਾ,adj,m,sg,,0,, '>
1.2 ਬੱਚਿਆਂ NN <fs af='ਬੱਚਾ,n,m,pl,3,o,, ' name='ਬੱਚਿਆਂ'>
))
2 (( NP <fs af='ਕਿਤਾਬ,n,f,sg,3,d,, ' head="ਕਿਤਾਬ">
2.1 ਕਿਤਾਬ NN <fs af='ਕਿਤਾਬ,n,f,sg,3,d,, ' name="ਕਿਤਾਬ">
))
3 (( VGF <fs af='ਪੜ੍ਹ,v,f,sg,any,, ,ਇਆ,ਇਆ' head="ਪੜ੍ਹੀ">
3.1 ਪੜ੍ਹੀ VM <fs af='ਪੜ੍ਹ,v,f,sg,any,, ,ਇਆ,ਇਆ' name="ਪੜ੍ਹੀ">
))
</Sentence>

```

**Figure 4.2: Output file generated by Punjabi shallow parser**

As shown in figure 4.2, the parser has identified three chunks, ‘ਛੋਟੇ ਬੱਚਿਆਂ ਨੇ’ ‘chōṭē bacciām nē’ ‘little children’, ਕਿਤਾਬ kitāb ‘book’ and ਪੜ੍ਹੀ paṛhī ‘read’. Here, ਛੋਟੇ chōṭē ‘little’ is an adjective with symbol ‘JJ’, ਬੱਚਿਆਂ bacciām ‘children’ and ਕਿਤਾਬ kitāb ‘book’ are nouns chunks with symbol ‘NN’ and ਪੜ੍ਹੀ paṛhī ‘read’ as finite verb chunk ‘VGF’.

Here, ‘fs’ is the feature structure which contains grammatical features of each word and ‘af’ is a composite attribute which consists of seven attributes namely root; lexical category; gender; number; person; case; tense, aspect and modality (TAM) information. In case, no value is given for a particular attribute the field is left blank. As shown in Figure 4.2, ਛੋਟੇ chōṭē ‘little’ has a root word ਛੋਟਾ chōṭā ‘little’; lexical category as ‘adj’ (adjective); gender as ‘m’ (male); number as ‘sg’ (singular) or ‘pl’ (plural), (*i.e.*, correct number is not resolved at this stage, so, both singular and plural forms are shown in the feature structure); person is ‘not applicable’ for this token; case is ‘o’ (oblique) and TAM is ‘not applicable’ for this token.

### Step 2: Extract root words

Root words are the smallest meaningful words which can’t be further divided. While playing sign animation only root words are considered. A java based system has been developed to extract root words and to provide the corresponding English words as shown in figure 4.3. The following are the root words extracted by the system for example sentence given in (4.1).

ਛੋਟਾ chōṭā ‘little’, ਬੱਚਾ baccā ‘child’, ਕਿਤਾਬ kitāb ‘book’, ਪੜ੍ਹਾ parh ‘read’

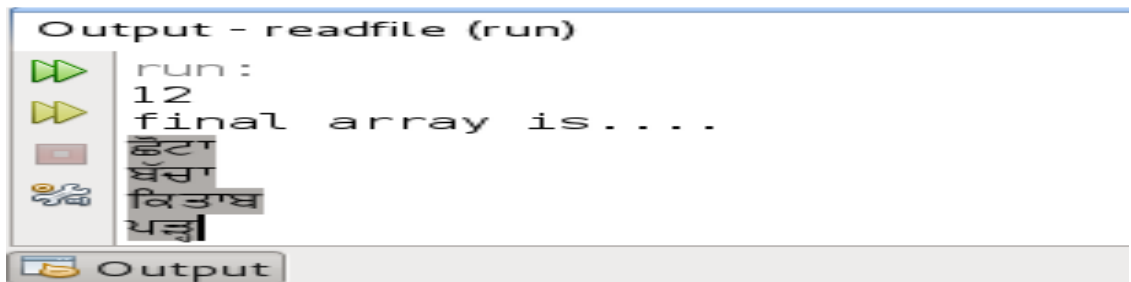


Figure 4.3: Extracted root words for example sentence given in (4.1)

### Step 3: Generate HamNoSys using eSIGNEditor and save its SiGML file

HamNoSys has been generated by eSIGNEditor which picks each word from lexicon and generates its corresponding HamNoSys as shown in figure 4.4.





### **4.3 Need for Creation of HamNoSys Generation System**

HamNoSys is based on Stokoe notation system. It is a symbolic representation of signs. Because of symbolic notations, it is not easy to understand it. It has around 200 different symbols. So, it is very difficult to remember all the HamNoSys symbols and their meanings. For this, sign language expert is required having complete knowledge of HamNoSys. But in India, it is very difficult to find such an expert. Hence, there is a need for GUI based HamNoSys generation system. In this thesis, GUI based HamNoSys generation system is developed. Anyone having little knowledge or no knowledge about HamNoSys can use this system to generate HamNoSys for different signs.

### **4.4 HamNoSys Generation System**

HamNoSys generation system generates HamNoSys for the Punjabi or English words. It takes Punjabi or English text as an input and generates HamNoSys for that word. HamNoSys symbols are categorization on the basis of hand shape, hand orientation, hand location and hand movement. User can directly select a particular symbol for generation of HamNoSys. When user enters a word for generating HamNoSys, system displays signed picture corresponding to inputted word. User can take idea from photograph, *i.e.*, which symbols will be required for generation of HamNoSys. When HamNoSys gets generated for a word then that word is stored into database along with its HamNoSys.

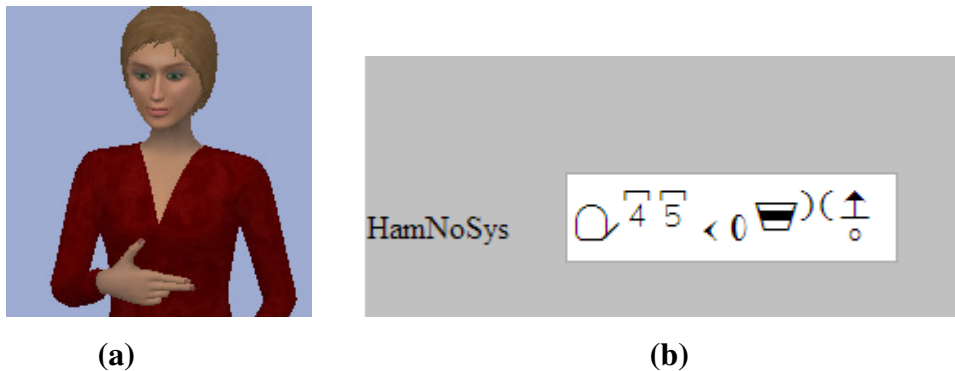
### **4.5 Implementation of HamNoSys Generation System**

HamNoSys generation system is a GUI based system developed in dot net language. In this system, all the HamNoSys symbols are defined. Symbols are categorized according to hand shapes, hand orientation, hand location and hand movements. A database is also maintained with it, which stores generated signs and their corresponding HamNoSys. There are around 350 different positions of body parts which are covered in this system for generating HamNoSys. The parameters of sign are categorized based on hand shapes, hand orientation, hand location and hand movements as shown in table 4.1. Numbers of HamNoSys symbols corresponding to parameters of signs are also shown in table 4.1. One symbol can generate many forms of sign after combining with other symbols.

**Table 4.1: Categorization of number of symbols covered in HamNoSys generation system**

Parameters of sign	Number of HamNoSys symbols
Hand Shapes	30
Hand orientations	26
Hand Locations	250
Hand Movements	24
Total	330

In HamNoSys generation system, when user enters a word for generating HamNoSys, system displays signed photograph corresponding to the entered word. From this photograph, user can easily guess which body parts are being used by the signer. From that user can decide what to choose from the menu options. There are 4 categories for user to choose. These are hand shapes, hand orientation, hand location and hand movements. Working of the system is explained step by step by generating HamNoSys for the word “EIGHT”.

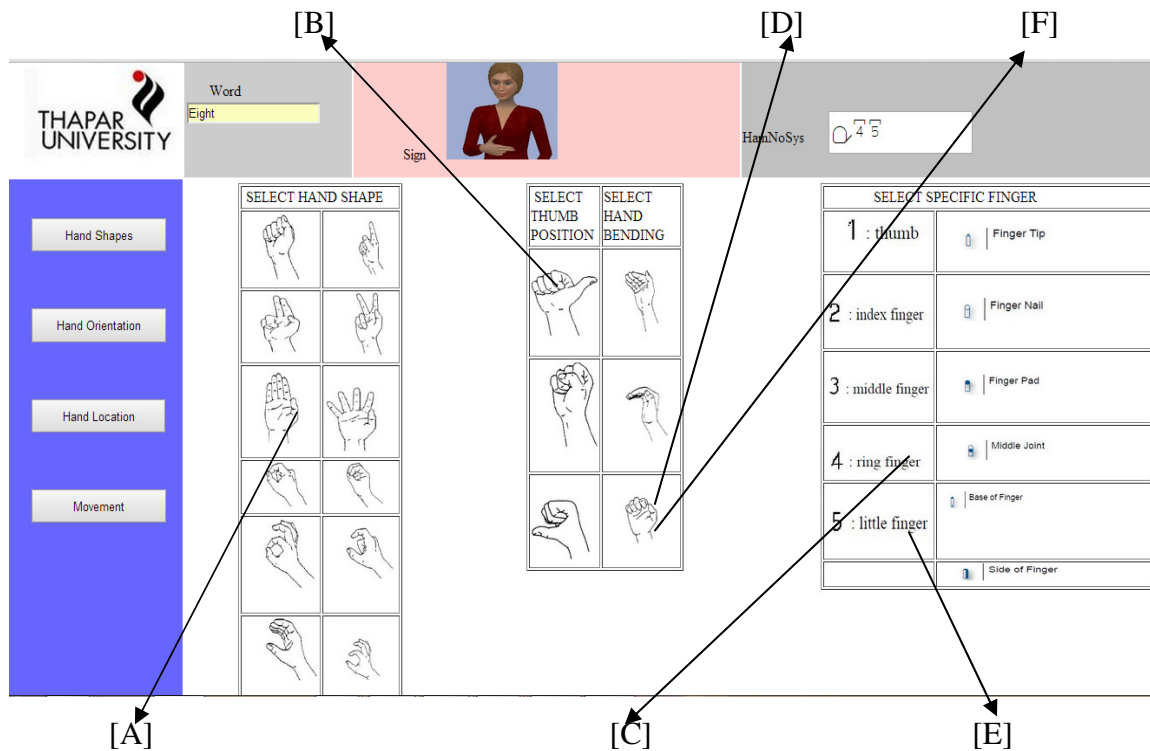


**Figure 4.8: (a) Represents sign for word “EIGHT” and (b) represents HamNoSys for word “EIGHT”**

#### 4.5.1 Representation of Hand Shapes

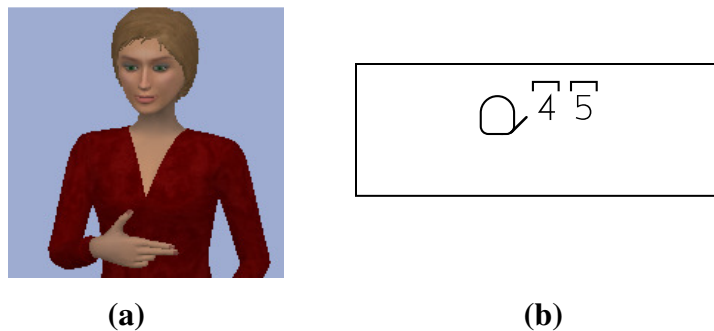
When user will select hand shapes, all the possible hand shapes will be displayed as shown in figure 4.9. It is must to select “hand shape” for using other menu options. After selecting hand shape, if there is a need of any type of hand bending or thumb position,

then that can be selected from “thumb position” or “hand bending”. If during signing, particular finger is playing role, then from “specific finger” that particular finger or finger location can be selected.



**Figure 4.9: Hand shapes for generating HamNoSys**

For generating sign animation of word “EIGHT”, firstly, hand shape is selected as shown in figure 4.9[A]. Then thumb position will be selected for hand shape as shown in figure 4.9[B]. After selecting hand shape, fingers and their bending will be selected as shown in figure 4.9[C], 4.9[D], 4.9[E] and 4.9[F] respectively. HamNoSys formed after selecting these for word “EIGHT” is shown in figure 4.10.



**Figure 4.10: (a) Represent sign for word “EIGHT” and (b) represents HamNoSys for word “EIGHT” after selecting hand shapes**

### 4.5.2 Representation of Hand Orientation

After selecting hand shape, there is requirement of hand orientations. Orientation is of two types, extended finger orientation and palm orientation. It decides the direction in which hand will be situated. First thing in this is extended finger orientation. It decides the direction of finger movement, *i.e.*, in which direction finger will move. User will decide according to his own position, *i.e.*, in which direction user is standing. Means, if user wants his hands in the space nearer to his body, then back view will be preferred. Here in back view, 8 basic finger orientations are defined. If user wants to select top view, means, user wants his hands away from his body. In right side view, both top and front views are combined at angle of 45°. User's hands will be away from his body at angle 45°. Palm orientation tells about the direction of palm. It decides whether palm will be towards the body, away from the body, up or down *etc.* There are 8 different positions of the palm. From these positions whatever position user will select, HamNoSys symbol corresponding to that will be generated in the textbox as shown in figure 4.11.

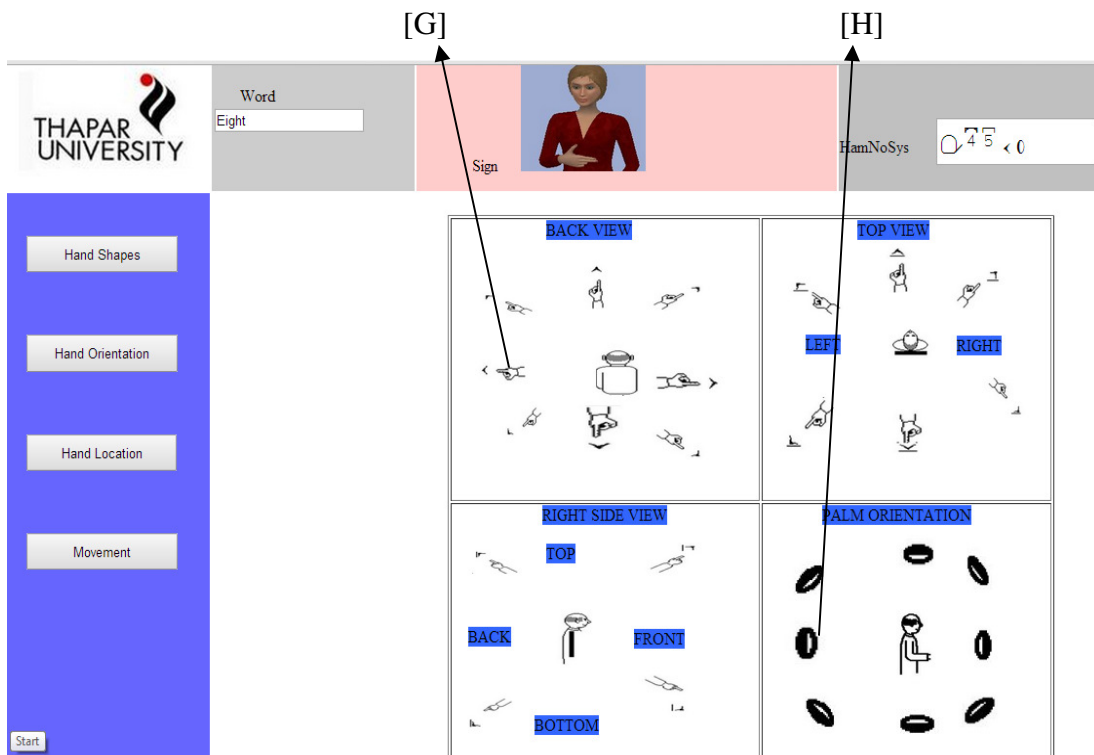
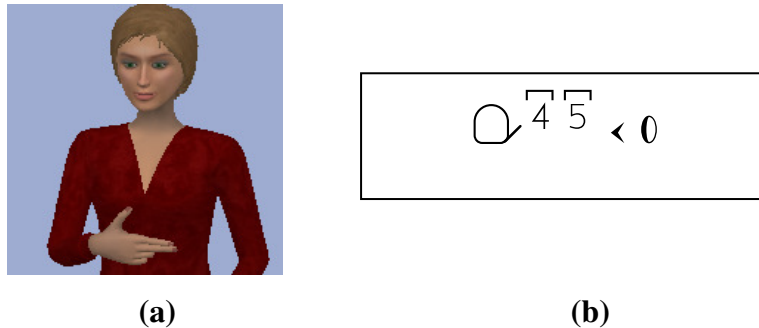


Figure 4.11: Hand orientation for generating HamNoSys

For generating sign animation of word “EIGHT”, after selecting hand shapes, hand orientation is selected. Here, firstly extended finger orientation is selected from back view as shown in figure 4.11[G]. Then palm orientation is selected as shown in figure 4.11[H]. HamNoSys formed after selecting hand orientation for word “EIGHT” is shown in figure 4.12.



**Figure 4.12: (a) Represent sign for word “EIGHT” and (b) Represents HamNoSys for word “EIGHT” after selecting hand orientation**

### 4.5.3 Representation of Hand Locations

After selecting orientation, user had created his hand shape. Now if there is a movement in hand then up to which body part our hand will move. This is decided by hand location. Hand location tells at which location our hand will stop. For example, in sign of word “woman” as shown in figure 1.7, location is near to nose. Locations decides which body part will be considered during signing. Basic hand locations are shown in figure 4.13.



**Figure 4.13: Hand Locations for generating HamNoSys**

In figure 4.13, main locations like head, pair locations, mouth, trunk, upper arm, lower arm and lower extremities are shown using buttons. When any location button from these will be selected, then detailed description corresponding to selected location will be opened in form of matrix as shown in figure 4.14.

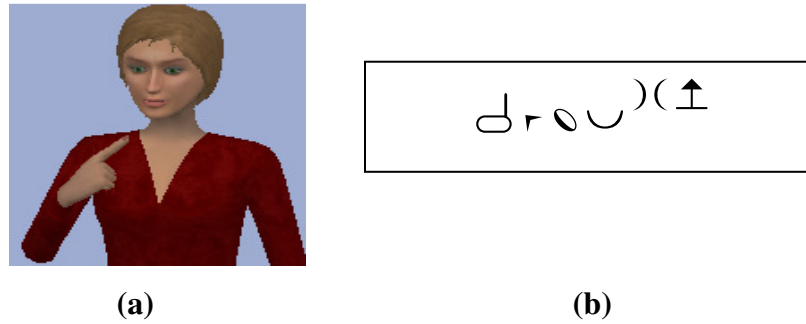
#### 4.5.3.1 Representation of HEAD locations

Description	Left To	Left Side of	Centre of	Right Side of	Right to
HEAD					
ABOVE THE HEAD					
FOREHEAD					
NOSE					
BELOW NOSE	NA				NA
MOUTH					
CHIN					
BELOW CHIN	NA				NA
NECK					

**Figure 4.14: Details of head locations**


In figure 4.14, all the locations which are related to head are covered. Here locations are head, above the head, forehead, nose, below nose, mouth, chin, below chin and neck. From these entire locations user can point towards for example left to nose, left side of nose, centre of nose, right side of nose or right to nose. All the possibilities that can occur by considering nose or any other head location are covered in figure 4.14. Details of location buttons shown in figure 4.13 are shown in figure 4.14, figure 4.16, figure 4.18, figure 4.20, figure 4.22 and figure 4.24 respectively. These figures give the details corresponding to mouth locations, Trunk locations, pair locations, upper arm locations and lower extremities respectively.

HEAD locations are not required for signing the word “EIGHT”, however for signing the word “SAY” this is required as shown in figure 4.14[I]. HamNoSys for word “SAY” is shown in figure 4.15.




**Figure 4.15: (a) Represent sign for word “SAY” and (b) Represents HamNoSys for word “SAY”**

#### 4.5.3.2 Representation of Locations related to MOUTH



Word




HamNoSys

Hand Shapes  
  
 Hand Orientation  
  
 Hand Location  
  
 Movement

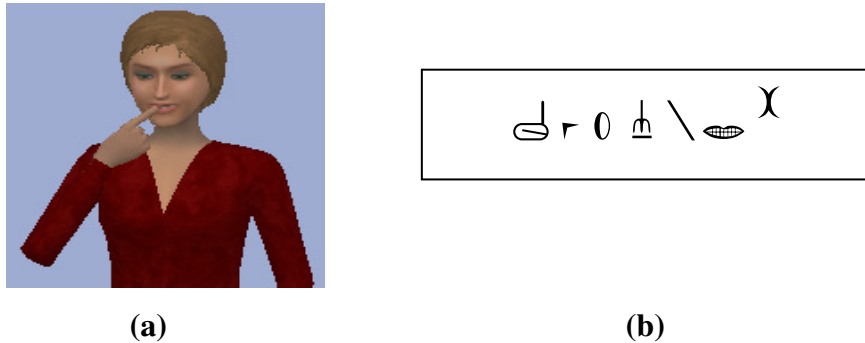
Description	Left To	Left Side of	Centre of	Right Side of	Right to
MOUTH					
TEETH	NA				NA
UPPER LIPS	NA				NA
UPPER TEETH ROW	NA				NA
UPPER SURFACE OF THE TONGUE	NA				NA
BLADE OF THE TONGUE					
LOWER SURFACE OF THE TONGUE	NA				NA
LOWER TEETH ROW	NA				NA
LOWER LIPS	NA				NA

[J]



**Figure 4.16: Details of Mouth locations**

MOUTH locations are not required to sign for the word “EIGHT”, however for signing the word “TEETH” this is required as shown in figure 4.16[J]. HamNoSys for word “TEETH” is shown in figure 4.17.



**Figure 4.17: (a) Represent sign for word “TEETH” and (b) Represents HamNoSys for word “TEETH”**

### 4.5.3.3 Representation of PAIRY LOCATIONS

Word

Sign

HamNoSys

Hand Shapes

Hand Orientation

Hand Location

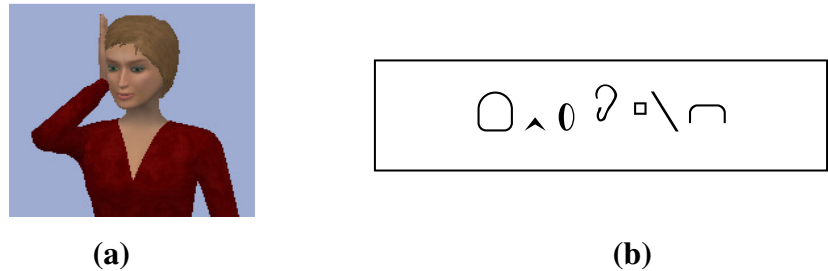
Movement

Description	LEFT TO THE LEFT	LEFT	BETWEEN THE TWO	RIGHT	RIGHT TO THE RIGHT
EYE BROWS					
UPPER EYE LIDS			NA		NA
EYES					
LOWER EYE LIDS	NA		NA		NA
EARS			NA		
EARLOBES			NA		
CHEEKS			NA		

**[K]**

**Figure 4.18: Details of Paired locations**

Paired locations are not required to sign for the word “EIGHT”, however for signing the word “SLEEP” this is required as shown in figure 4.18[K]. HamNoSys for word “SLEEP” is shown in figure 4.19.



**Figure 4.19: (a) Represent sign for word “SLEEP” and (b) Represents HamNoSys for word “SLEEP”**

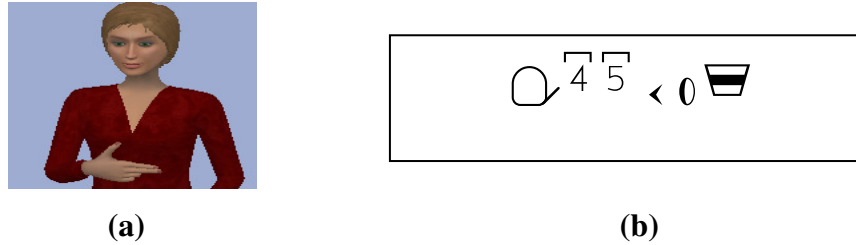
#### 4.5.3.4 Representation of TRUNK locations

The screenshot shows a software interface for selecting trunk locations. At the top, there is a header with the Thapar University logo, a "Word" field containing "Eight", a "Sign" field with a 3D figure, and a "HamNoSys" field containing a sequence of symbols. Below the header is a sidebar with buttons for "Hand Shapes", "Hand Orientation", "Hand Location", and "Movement". The main area contains a table of trunk locations. An arrow labeled "[L]" points to the "Centre of" column in the "BREAST LINE" row.

Description	Left To	Left Side of	Centre of	Right Side of	Right to
TOP OF SHOULDER	<input type="checkbox"/>	<input type="checkbox"/>	NA	<input type="checkbox"/>	<input type="checkbox"/>
SHOULDER LINE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
BREAST LINE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
BELLY LINE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
ABDOMINAL LINE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>

**Figure 4.20: Details of Trunk locations**

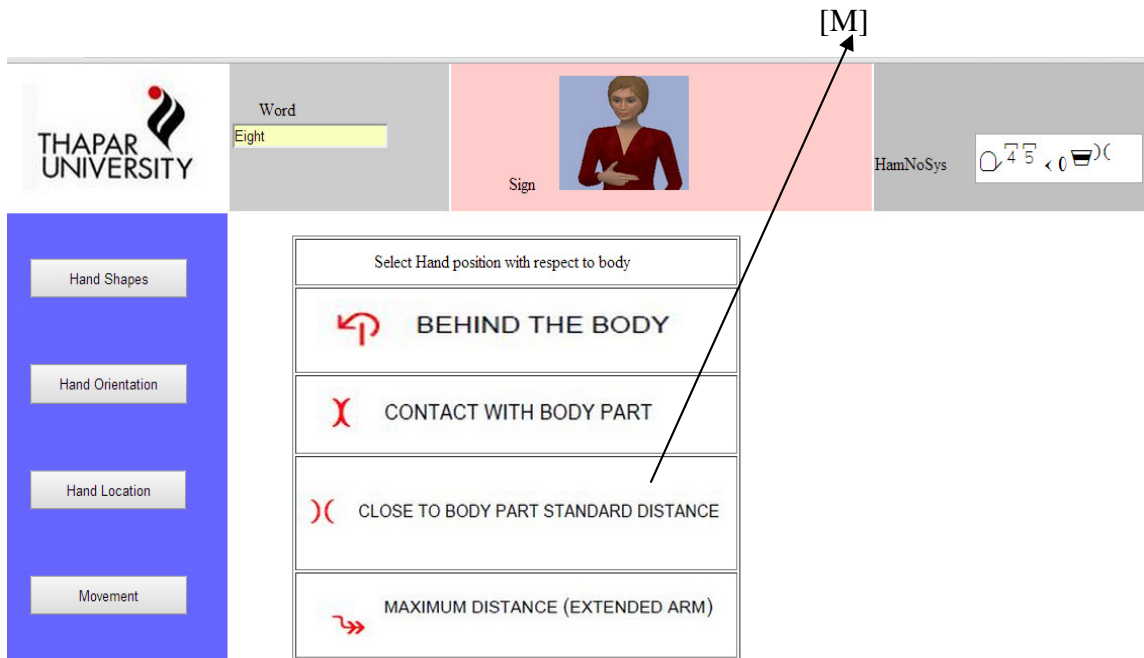
For generating sign animation of word “EIGHT”, after selecting hand orientation hand location is selected from Trunk locations. Location selected is in centre of breast line as shown in figure 4.20[L]. HamNoSys formed after selecting hand location for word “EIGHT” is shown in figure 4.21.



**Figure 4.21: (a) Represent sign for word “EIGHT” and (b) Represents HamNoSys for word “EIGHT” after selecting hand location**

#### 4.5.3.5 Representation of Hand positions with respect to the body

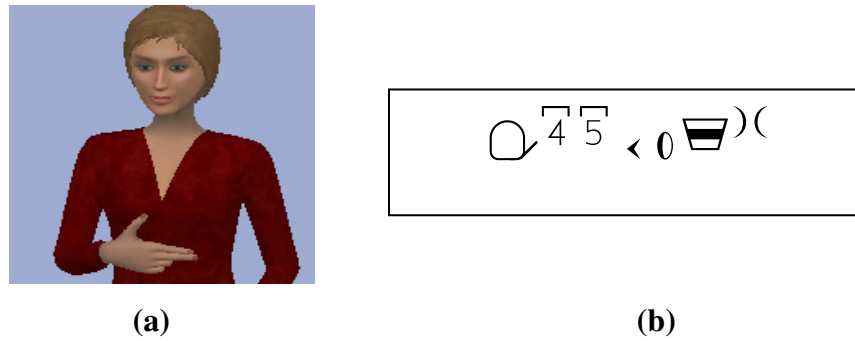
After selecting particular location there is a requirement to select the hand position with respect to the selected body part. Basic Hand positions are shown in figure 4.22.



**Figure 4.22: Hand position with respect to selected location**

In figure 4.22, if hand will be in contact with the body then symbol “contact with body part” will be selected. It displays whether hand is in contact, behind, close or at maximum distance from the body.

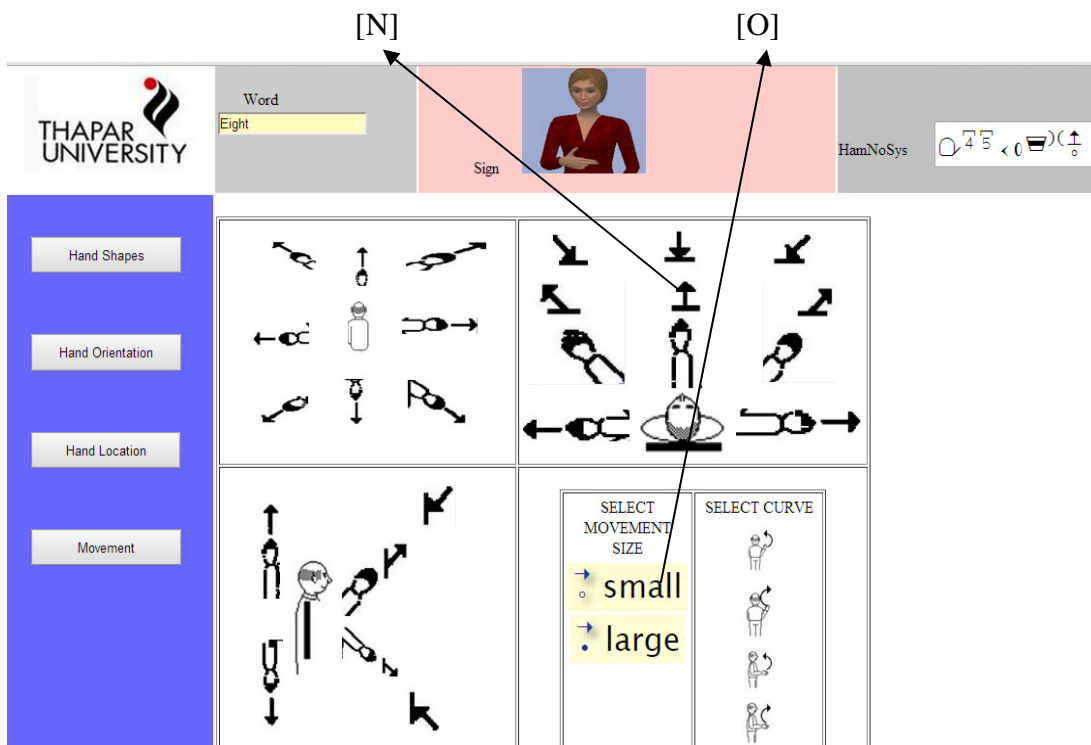
For generating sign animation of word “EIGHT”, after selecting hand location hand contact with respect to the body is selected. Hand is close to the body with standard distance as shown in figure 4.22[M]. HamNoSys formed after selecting hand contact with the body for word “EIGHT” is shown in figure 4.23.



**Figure 4.23: (a) Represent sign for word “EIGHT” and (b) Represents HamNoSys for word “EIGHT” after selecting hand contact with body**

#### 4.5.4 Representation of Hand Movements

If hand moves during signing, then user will select movement button from menu options. Two types of movements, straight and curved movements are shown in figure 4.24. If movement is straight then no need to select curved movement. User will select only the straight movement according to his body position. In case of curved movements, user will select one curve from four curved options.

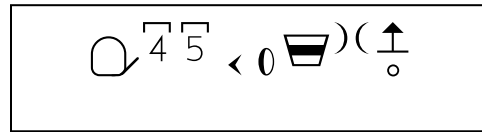


**Figure 4.24: Hand movements for generating HamNoSys**

For generating sign animation of word “EIGHT”, after selecting hand contact with the body hand movement is selected. Here, hand movement is selected from the top view as shown in figure 4.24[N]. After selecting basic movement, size of movement is selected as shown in figure 4.24[O]. HamNoSys formed after selecting hand movement for word “EIGHT” is shown in figure 4.25.



(a)



(b)

**Figure 4.25: (a) Represent sign for word “EIGHT” and (b) Represents HamNoSys for word “EIGHT” after selecting hand movements**

HamNoSys for word “EIGHT” has been generated. It can be saved into database.

#### 4.6 Description of System by Generating HamNoSys for word “TWO”

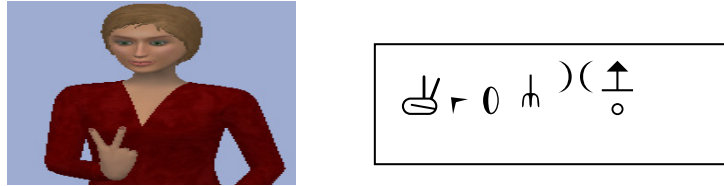


Figure 4.26: Sign and HamNoSys for word “TWO”

Table 4.2: Steps to generate HamNoSys for word “TWO”

Sequence of Action	Referred in figure	Corresponding HamNoSys
Hand Shape	3.29(a) [A]	ʌ
Hand bending	3.29(a) [B]	ʌ
Hand orientation	3.29(b) [C]	r
Palm Orientation	3.29(b) [D]	0
Hand Location	3.29(c) [E]	ʎ
Hand contact with body	3.29(d) [F]	) (
Hand Movement	3.29(e) [G]	ʌ̇
Movement size	3.29(e) [H]	ʌ̇ ○

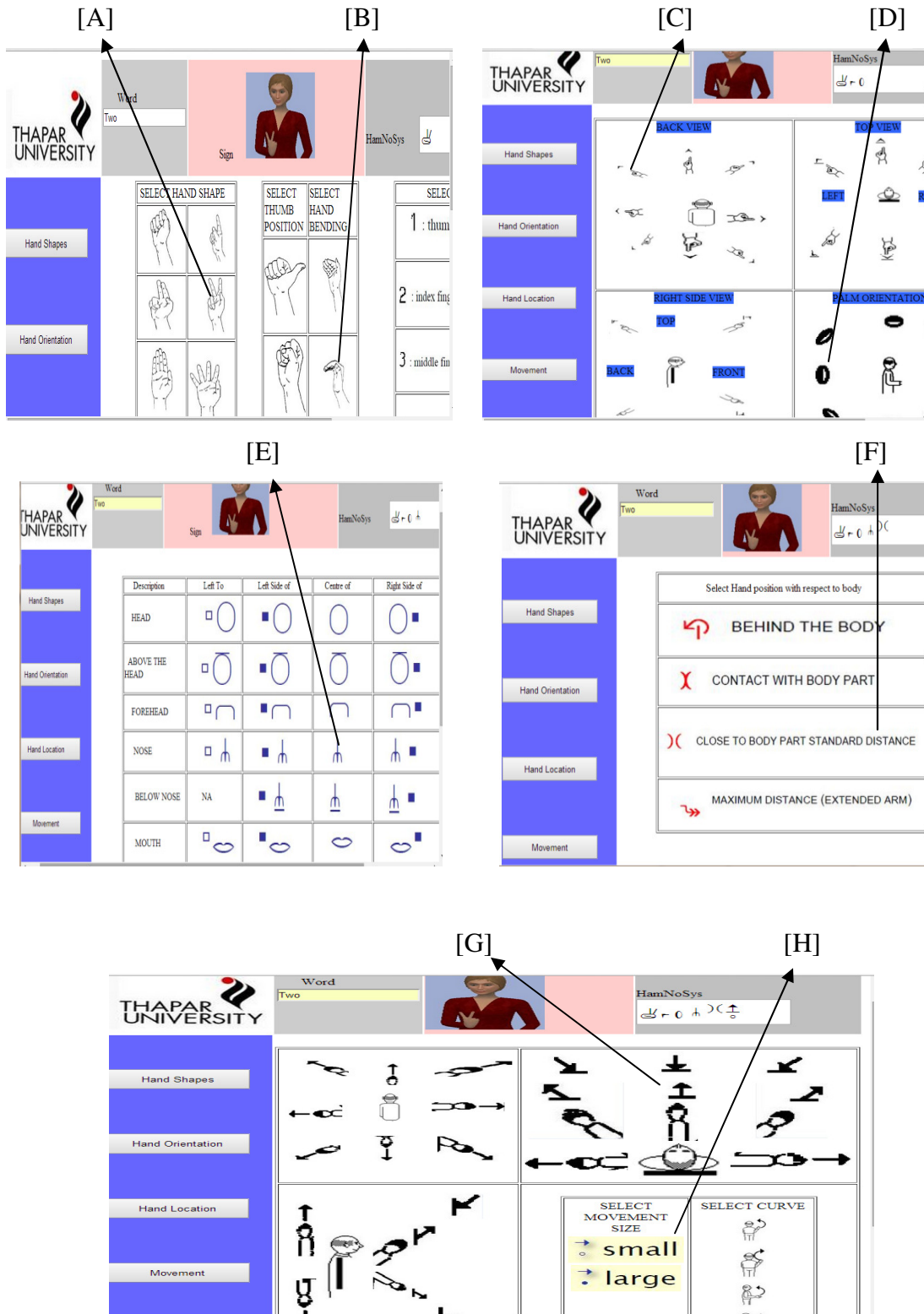


Figure 4.27: Steps of HamNoSys generation for word “TWO”

## **Chapter Summary**

In this chapter, sign language generation system for Punjabi language has been proposed. It takes Punjabi sentence as input and play sign animation for it. For performing animation, it takes Punjabi sentence as input and using Punjabi shallow parser it parses the sentence into tokens having root words. These root words are extracted using java program. Then HamNoSys is generated for these root words. HamNoSys is further converted to SiGML which plays sign animation for the sentence. For generating HamNoSys, a HamNoSys generation system has been developed. It generates HamNoSys for word and stores word along with its HamNoSys in its database. Whenever sign language generation system for Punjabi language requires HamNoSys for any word then it can map word and pick generated HamNoSys from the database of HamNoSys generation system.

5.1 Testing of Sign Language generation system for Punjabi language

Sign Language generation system for Punjabi language has been tested for 200 simple Punjabi sentences. Some of example sentences used for testing and their corresponding sign generated by Punjabi sign language generation system are illustrated in table 5.1. The quality of this system is evaluated manually by the sign language experts and the results of the system are very encouraging.

Table 5.1: Example sentences used for testing of Punjabi sign language generation system

Input Text	HamNoSys	Sign Animation
<p>Punjabi sentence: ਮੈਨੂੰ ਸੌ ਰੁਪਏ ਦਿਉ।</p> <p>Transliterated Punjabi sentence: mainūṃ sau ruṃpāē diu</p> <p>Equivalent English sentence: Give me a hundred rupees.</p>		<p>ਮੈਨੂੰ ਦਿਉ      ਸੌ      ਰੁਪਏ Give me      hundred      rupees</p>
<p>Punjabi sentence: ਤੁਸੀਂ ਮੈਨੂੰ ਸਲਾਹ ਦਿਉਗੇ।</p> <p>Transliterated Punjabi sentence: tusīṃ mainūṃ salāh diugē</p> <p>Equivalent English sentence: You will give me advice.</p>		<p>ਤੁਸੀਂ      ਮੈਨੂੰ      ਸਲਾਹ      ਦਿਉਗੇ You      give me      advice      will</p>

Input Text	HamNoSys	Sign Animation
<p>Punjabi sentence: ਅਸੀਂ ਸਰਵਿਸ ਦਿੰਦੇ ਹਾਂ।</p> <p>Translatered Punjabi sentence: asīṃ sarvis dindē hām</p> <p>Equivalent English sentence: We provide services.</p>		<p>ਅਸੀਂ                      ਸਰਵਿਸ                      ਦਿੰਦੇ</p> <p>We                                      services                                      provide</p>
<p>Punjabi sentence: ਕ੍ਰਿਪਾ ਕਰਕੇ ਮੈਨੂੰ ਮਾਫ਼ ਕਰ ਦਿਉ।</p> <p>Translatered Punjabi sentence: kripā karkē mainūṃ māf kar diu</p> <p>Equivalent English sentence: Please forgive me.</p>		<p>ਕ੍ਰਿਪਾ ਕਰਕੇ                      ਮੈਨੂੰ                      ਮਾਫ਼</p> <p>Please                                      me                                      forgive</p>
<p>Punjabi sentence: ਉਹ ਫੁਟਬਾਲ ਖੇਡਦੇ ਹਨ।</p> <p>Translatered Punjabi sentence: uh phuṭbāl khēḍdē han</p> <p>Equivalent English sentence: They play football.</p>		<p>ਉਹ                                      ਫੁਟਬਾਲ                                      ਖੇਡ</p> <p>They                                      football                                      play</p>


















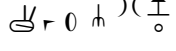

















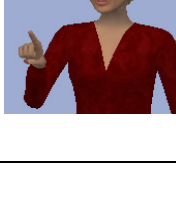
## 5.2 Testing of HamNoSys Generation System








HamNoSys generation system has been tested on 200 words. Some of example words used for testing and their corresponding HamNoSys generated by system are illustrated in table 5.2.




**Table 5.2: Words used for testing and their corresponding HamNoSys generated by HamNoSys generation system**

Sr. No.	Inputted word	Photograph	Sequence of Actions				Generated HamNoSys
			Hand shapes	Orient ation	Location	Move ment	
1.	Jealous			r 0	☐)(	→	
2.	Deaf			r \ r 0	? X	NA	
3.	Later			r 0	☐	↑ c	
4.	Say			r 0	☐)(	↑	
5.	That			h 0	☐	↘	
6.	Two			r 0	h )(	↑ ○	

Sr. no.	Inputted Word	Photograph	Sequence of Actions				Generated HamNoSys
			Hand Shape	Orient ation	Location	Move ment	
7.	Thumb		☞	< 0	☞)(	NA	☞ < 0 ☞)(
8.	Stubborn		☞	< 0	☞)(	↓	☞ < 0 ☞)(↓
9.	Sleep		☞	^ 0	? □ \ ㄩ	NA	☞ ^ 0 ? □ \ ㄩ
10.	Bed		☞	^ 0	○ \ 2 ) (	NA	☞ ^ 0 ○ \ 2 ) (
11.	Discuss		☞	^ 0	☞)(	↘	☞ ^ 0 ☞)(↘
12.	Child		☞	△ 0	☞	↓	☞ △ 0 ☞ ↓
13.	Pick		☞	△ 0	☞	NA	☞ △ 0 ☞

Sr. no.	Inputted Word	Photograph	Sequence of Actions				Generated HamNoSys
			Hand Shape	Orient ation	Location	Move ment	
14.	Me		☞	└0	☞χ	NA	☞└0☞χ
15.	You		☞	△\└0 ∖o	☞	NA	☞△\└0\o☞
16.	Give me		☞	<o	NA	↓	☞<o↓
17.	Mobile hold		☞	△o\o	☞▪	↗	☞△o\o☞▪↗
18.	Mobile		☞	△o\o	☞▪	NA	☞△o\o☞▪
19.	Boy		☞	^0	∪▪	←	☞^0∪▪←
20.	But		☞	△\^o	☞▪	↑	☞△\^o☞▪↑

Sr. no	Inputted Word	Photograph	Sequence of Actions				Generated HamNoSys
			Hand Shape	Orient ation	Location	Move ment	
21.	Medical		∩	< 0	☐ \ ☐ ▪	← ☐ ☐ \ ☐	∩ < 0 ☐ \ ☐ ▪ ← ☐  ☐ \ ☐
22.	Told		☐	r 0	☐ ) (	↑	☐ r 0 ☐ ) ( ↑
23.	Down		☐	l 0	☐	☐	☐ l 0 ☐ ☐
24.	Word		☐	> \ r 0	☐ ) (	↑	☐ > \ r 0 ☐ ) ( ↑
25.	Hand		☐ \ ☐	r 0	NA	☐	☐ \ ☐ r 0 ☐
26.	Lovely		☐	^ \ Δ 0	☐ \ ☐ ) (	☐	☐ ^ \ Δ 0 ☐ \ ☐  ) (☐
27.	Teeth		☐	r 0	☐ \ ☐ X	NA	☐ r 0 ☐ \ ☐ X

Sr. no.	Inputted Word	Photograph	Sequence of Actions				Generated HamNoSys
			Hand Shape	Orient ation	Location	Move ment	
28.	Strange		☞	< 0	☪ ☐ ) (	☞ ○	☞ < 0 ☪ ☐ ) ( ☞ ○
29.	Like		☞	< 0	☪ X	☞ ○ ○	☞ < 0 ☪ X ☞ ○ ○
30.	Eight		☞ <sup>4</sup> 5	< 0	☪ ) (	☞ ○	☞ <sup>4</sup> 5 < 0 ☪ ) ( ☞ ○

## Chapter 6

### Conclusion and Future Scope

---

#### 6.1 Conclusion

Deaf people or people who take birth in deaf families learn sign language as their first language. Written language becomes their second priority. These people prefer to access information in form of sign language only. Providing information in form of video clips is very expensive so sign animation is a promising approach. For generating sign animation, sign language generation system for Punjabi language is proposed in this thesis. It generates signs for Punjabi sentences. For performing sign animation, it takes Punjabi sentence as input. Then using Punjabi shallow parser, it parses the sentence into its root words and save them in a file. These root words are extracted from file using java program. Then HamNoSys is generated corresponding to these root words. HamNoSys is further converted into XML form known as SiGML. Virtual human in tool JA SiGML URL APP takes input of SiGML and generates signs corresponding to words given in the sentence.

HamNoSys is necessary to generate sign animation. It is a Stokoe based notation system having around 200 different symbols. For remembering and understanding these symbols, sign language expert is required. But it is difficult to find such an expert. So, HamNoSys generation system is developed in this thesis. It is a GUI based system in which all the HamNoSys symbols are defined. HamNoSys symbols are categorized in it based on hand shape, hand orientation, hand location and hand movements. It covers around 350 HamNoSys symbols. HamNoSys can be generated using these symbols corresponding to any word. After generation of HamNoSys it stores word along with its HamNoSys into database. Sign language generation system for Punjabi language uses this stored HamNoSys and generates sign animation corresponding to it. These systems are tested on around 200 simple sentences and results of systems are very encouraging.

## **6.2 Limitations and Future Scope**

Sign language generation system for Punjabi language works on only simple Punjabi sentences. It can be extended to work with more complex sentences. Currently this system makes use of different tools. A complete system can be developed which can collaborate all these tools into one system. HamNoSys generation system includes basic hand movements. It can be extended to generate all the possible hand movements and non-manual expressions. This system can be made complete web based means anybody on internet can use it. A mobile application can also be developed for sign language generation from text.

## References

---

- [1] Tirthankar, D., Sambit, S., Sandeep, K., Synny, D., and Anupam, B., 2008. “A *Multilingual Multimedia Indian Sign Language Dictionary Tool*”. In Sixth workshop on Asian Language Resources, India: Indian School of Business, Hyderabad, pp. 11-12
- [2] Zeshan U., Madan M., Vasishta, and Sethna M., 2004. “*Implementation of indian sign language in educational settings*”, Asia Pacific Disability Rehabilitation Journal, Volume 15, Number 2, pp. 15-35
- [3] Vasishta M., Woodward J., and DeSantis S., 1998. “*An Introduction to Indian Sign Language*”, All India Federation of the Deaf, Third Edition.
- [4] Zeshan U., 2003. “*Indo-Pakistani Sign Language Grammar: A Typological Outline*”, Sign Language Studies, Volume 3, Number 2, pp. 157-212
- [5] Stokoe W. C., 1960. “*Sign language structure: an outline of the visual communication systems of the American deaf*”, 2nd edition, 1978. Silver Spring, MD: Linstok Press.
- [6] Kennaway, R., 2001. “*Synthetic animation of deaf signing gestures*”. In International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction, GW 2001, London, UK, Springer, pp. 149–174
- [7] Sara, M., Harold, S., Robert, S., Shane, G., and Sandipan, D., 2010. “*Building a Sign Language corpus for use in Machine Translation*”, In 4th Workshop on the Representation and Processing of Sign Languages, Corpora and Sign Language Technologies, Dublin City University, Ireland, pp. 172-177
- [8] Prillwitz, S., and Schulmeister, R., 1987. “*HamNoSys. Hamburg Notation System for Sign Languages. An introduction*”, Hamburg, Germany.
- [9] Hanke, T., 2001. “*Sign language transcription with syncWRITER*”, Sign Language and Linguistics 4, pp. 267–275
- [10] Prillwitz, S., Leven, R., Zienert, H., Hanke, T., and Henning, J., 1989. “*HamNoSys Version 2.0: Hamburg Notation System for Sign Languages: An Introductory Guide*”, International Studies on Sign Language and Communication of the Deaf, Volume 5, Signum Press, Hamburg, Germany.
- [11] Valli, C., and Lucas, C., 2000. “*Linguistics of American Sign Language*”, 3rd edition, Washington, DC: Gallaudet University Press.

- [12] Robert, S., 2010. “*HamNoSys4.0 For Irish Sign language*”, Workshop Hand book, Centre for Next Generation Localization, Version 3.0, Draft 2.0, Dublin City University, Ireland, pp. 1-65
- [13] Kennaway, R., 2003. “*Experience with and requirements for a gesture description language for synthetic animation*”. In 5th International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, GW2003, Genova, Italy, Springer, pp. 300–311
- [14] Elliott, R., Glauert, J.R.W., Jennings, V., and Kennaway, J.R., 2004. “*An overview of the SiGML notation and SiGML Signing software system*”, In proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, Lisbon, pp. 98–104
- [15] Kennaway, R., 2001. “*Synthetic animation of deaf signing gestures*”. In International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction, GW 2001, London, UK, Springer, pp. 149–174
- [16] Huenerfauth, M., 2003. “*A Survey and Critique of American Sign Language Natural Language Generation and Machine Translation Systems*”. Technical Report, Computer and Information Science, University of Pennsylvania, MS-CIS-03-32
- [17] Tirthankar, D., Sandipan, D., Anupam B., 2008. “*Prototype Machine Translation System From Text-To-Indian Sign Language*”, In IJCNLP-08 Workshop on NLP for Less Privileged Languages, Asian Federation of Natural Language Processing, Hyderabad, India, pp. 19–26
- [18] Speers A., 2001. “*Representation of American Sign Language for Machine Translation*”, PhD Dissertation, Department of Linguistics, Georgetown University.
- [19] Cox S., Lincoln M., Tryggvason J., Nakisa M., Wells M., Tutt M., and Abbott S., 2002. “*Tessa, a system to aid communication with deaf people*”, In Proceedings of the Fifth international ACM conference on Assistive technologies.
- [20] Grieve-Smith, Angus B., 2002. “*SignSynth: A Sign Language Synthesis Application Using Web3D and Perl*”, Linguistic Department, Humanities 526, The University of New Mexico Albuquerque, NM 87131 USA, Springer pp. 134-145
- [21] Zhao, L., Kipper, K., Schuler, W., Vogler, C., Badler, N., and Palmer, M., 2000. “*A Machine Translation System from English to American Sign Language*”, Association for Machine Translation in the Americas.

- [22] Sáfár, E., and Marshall, I., 2001. “*The architecture of an English-text-to-Sign-Languages translation system*”, In Recent Advances in Natural Language Processing (RANLP), Tzigris Chark, Bulgaria, pp. 223-228
- [23] Veale, T., Conway, A., and Collins, B., 1998. “*The challenges of cross-modal translation: English to sign language translation in the ZARDOZ system*”. In Machine Translation 13, pp. 81-106
- [24] Kaplan, RM., 1989. “*The formal architecture of lexical-functional grammar*”, Journal of Information Science and Engineering 5, pp. 305-322
- [25] Sleator, D., and Temperley, D., 1991. “*Parsing English with a Link Grammar*”, Technical report, Computer Science, Carnegie Mellon University, CMU-CS-91-196
- [26] Kamp, H., and Reyle, U., 1993. “*From Discourse to Logic Introduction to Model theoretic Semantics of Natural Language*”, Formal Logic and Discourse Representation Theory, Kluwer Academic Publishers, Dordrecht.
- [27] Pollard, C. J., and Sag, I. A., 1994. “*Head-driven Phrase Structure Grammar*”, University of Chicago Press, Chicago, IL.
- [28] Wilcox, S., Scheibman, J., Wood, D., Cokely, D., and Stokoe, W.C., 1994. “*Multimedia dictionary of American Sign Language*”. In Proceedings of the first annual ACM conference on Assistive technologies, ACM Press, New York, NY, USA, pp. 9–16
- [29] Geitz, S., Hanson, T., and Maher, S., 1996. “*Computer generated 3-dimensional models of manual alphabet hand shapes for the World Wide Web*”. In Proceedings of the second annual ACM conference on Assistive technologies, ACM Press, New York, NY, USA, pp. 27–31
- [30] Solanki, K., 2013. “*Indian Sign Languages using Flex Sensor Glove*”, International Journal of Engineering Trends and Technology (IJETT), vol. 4, ISSN: 2231-5381, pp. 2478-2480
- [31] Hanke, T., and Hortensia, P., 2003. “*Essential sign language information on government networks*”. Intelligent Sign Editor, EDC-22124, pp. 1-27
- [32] Hanke, T., 2004. “*HamNoSys – Representing Sign Language Data in Language Resources and Language Processing Contexts*”, University of Hamburg, Binderstraße 34, 20146 Hamburg, Germany.

- [33] Parteek, K., 2012. “*UNL Based Machine translation System for Punjabi Language*”, PhD Dissertation, Department of Computer Science and Engineering, Thapar University, Patiala.
- [34] Bharati, A., Sangal, R., and Sharma, D., 2007. “*SSF: Shakti Standard Format Guide*”, Technical Report, Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India, pp. 1-25
- [35] Bharati, A., and Sangal, R., 1993. “*Parsing free word order languages in the Paninian framework*”, In Proceedings of the Annual Meeting of Association for Computational Linguistics, New Jersey, pp. 105-111

## List of Publications

---

### Research Paper Accepted

- Rupinder Kaur and Parteek Kumar, “HamNoSys Generation System for Sign Language” in *International Conference on Advances in Computing, Communications and Informatics, ICACCI Track 1: Advanced Computing*, ICACCI-2014, Delhi, India

### Research Paper under Review

- Rupinder Kaur and Parteek Kumar, “Sign Language Generation System for Punjabi Language” in *Sadhana - Academy Proceedings in Engineering Science*, Journal of Springer, SADH-D-14-00190 (SCI Indexed).
- Rupinder Kaur and Parteek Kumar, “Survey on Sign Language Generation Systems” in *Bahri Publications*, October 2014.