

# ARCHITECTURAL MODEL FOR TEXT TO SIGN LANGUAGE INTERPRETER(TSLI)

Liyana Pathirannehelage Manupriya Banu Liyanapathirana

(199380R)

The thesis submitted in partial fulfillment of the requirements for the  
degree Master of Science in computer science specializing in Software  
Architecture



Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

June 2022

To My Parents.

## Declaration

*I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.*

*Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future work (such as articles or books).*

*Name of Candidate: L.P.M.B.Liyanapathirana*

*Signature: ..... Date: June 26, 2022*

*The above candidate has carried out research for the Masters thesis under my supervision.*

*Name of the supervisor: Prof. Indika Perera*

*Signature: ..... Date: June 26, 2022*

## Abstract

The deaf and hearing-impaired make up a considerable percentage of the world community having some essential needs that researchers have recently begun to target. There is no such proper way to communicate with deaf society instead of learning sign language. Sign language is a natural language that can be used as a powerful weapon for communicating with deaf society. The target objective of the research is to propose a Generic Architectural model to bridge the communication gap between deaf and non-deaf people. In the desertion, this problem was tackled by presenting the "Architectural model for Text to Sign Language Interpreter" system to translate text into any given Sign Language. Language parser Handler service is introduced as an API Contract. This Language parser Handler service acts as an abstraction layer for the backend services which are doing the NLP processing. Any researcher interested in translating sign language from a text can use this proposed model. A new language can be introduced to the model with minimum configurations. Instead of developing the entire model they just need to focus on the actual Language processing part. It should be a RESTful microservice. This model can be used to test their API

In a nutshell, the Language-specific parser service does the following. First, the individual words are extracted by tokenizing the input sentence. The morphological analysis is then performed to identify the basic elements of the respective words. Basic components can be a sign stem, sign marker, or fingerspelling characters. After that, the identified basic word components (stream of tokens) are sent back to the client-side as a JSON string.

The primary tokens are extracted from the JSON output. Then the tokens are mapped with the corresponding SiGML files which are forwarded to the signing avatar embedded in the client UI. The final output is made by the signing avatar model which gives a stream of sign frames.

keywords: Sign Language, Service-Oriented Architecture

## Acknowledgments

I would like to thank and extend my heartfelt gratitude to Prof.Indika Perera for the continuous supervision of my research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me throughout the research, and in writing this dissertation

Besides my supervisor, I would like to thank Mrs.Renuka Padmalatha who has given me knowledge of the Sinhala sign language, for her encouragement, insightful comments, and time consecrate for the research.

Special thanks to all the people who had done their research based on related areas and had committed their lives for the success of inventions. Without their contribution, this thesis would not be a reality.

My sincere thanks also go to my parents, my wife, and my family who had given me help and courage to do this research and to be strong in life as a successor. Also, I thank my friends who are always with me in any situation.

In order to preserve anonymity they cannot be named. Without their help, carrying out this research would not be possible.

Manupriya Banu Liyanapathirana, June 26, 2022

## Table of Contents

<b>Declaration</b>	<b>iii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>List of Figures</b>	<b>x</b>
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Aim and Objectives . . . . .	4
1.3 Scope . . . . .	5
1.4 Structure of the thesis . . . . .	7
<b>CHAPTER 2: LITERATURE REVIEW</b>	<b>8</b>
2.1 Deaf culture . . . . .	8
2.2 Sign Language . . . . .	9
2.2.1 Sinhala Sign Language . . . . .	10
2.2.2 Sign language features . . . . .	11
2.2.2.1 Sign Order . . . . .	11
2.2.2.2 Signing Space, Placement and Pronouns . . . . .	12
2.2.2.3 Directional or Agreement Verbs . . . . .	12
2.2.2.4 Classifiers . . . . .	12
2.2.2.5 Time Lines . . . . .	13
2.2.3 Sign Language Phonology . . . . .	13
2.2.4 Sign Alphabet . . . . .	15

2.3	Existing systems . . . . .	19
2.3.1	Speech to Sign Language Interpreter System (SSLIS) . . . . .	19
2.3.2	Speech to Sign Language translation system for Spanish . . . . .	20
2.3.3	Existing 3D avatar tools . . . . .	21
2.3.3.1	eSign . . . . .	22
2.3.3.2	ViSiCAST . . . . .	23
2.3.3.3	JASigning . . . . .	24
2.3.3.4	CWA Signing Avatars . . . . .	24
2.4	Technologies . . . . .	25
2.4.1	Python . . . . .	25
2.4.2	Flask framework . . . . .	26
2.4.2.1	Installing python packages with Flask . . . . .	26
2.4.2.2	Routes and View Functions . . . . .	26
2.4.2.3	Server Startup . . . . .	27
2.4.3	NoSQL Databases . . . . .	27
2.4.3.1	Existing NoSQL Solutions . . . . .	28
2.5	Software Architecture Evaluation Techniques . . . . .	29
2.5.1	ATAM - Architecture Trad-off Analysis method . . . . .	30

**CHAPTER 3: METHODOLOGY 34**

3.1	Research methodology . . . . .	34
3.2	Prerecorded videos and avatar library . . . . .	35
3.3	System Design . . . . .	35
3.3.1	UI Client . . . . .	37
3.3.2	API Gateway . . . . .	37
3.3.3	Language Parser Handling Service . . . . .	38
3.3.4	Language parser services . . . . .	41
3.3.5	Client Event Handler . . . . .	43
3.3.6	SiGML converter . . . . .	45



<b>CHAPTER 4: Evaluation of the Architecture</b>	<b>48</b>
4.1 Quality Attributes . . . . .	48
4.2 ATAM - Architecture Trad-off Analysis method . . . . .	52
4.3 System comparison with an existing application . . . . .	59
4.3.1 Evaluation procedure . . . . .	59
4.3.2 Result analysis . . . . .	60
 <b>CHAPTER 5: Conclusion</b>	 <b>64</b>
5.1 Discussion . . . . .	66
5.2 Future Work . . . . .	67
 <b>References</b>	 <b>68</b>

## List of Figures

1.1	High level Architecture of the proposed system . . . . .	5
2.1	Corresponding sign sequence for <i>Who are your friends</i> [24] . . . . .	11
2.2	Basic categorization of Sign Language . . . . .	14
2.3	Sinhala sign alphabet . . . . .	16
2.4	Minor vowel signs for sound <i>ai</i> and <i>au</i> . . . . .	17
2.5	Phoneme modifiers in SSL . . . . .	17
2.6	Different letter pairs in SSL . . . . .	18
2.7	Finger-spelled signs for word “ <i>SRI LANKA</i> ” . . . . .	18
2.8	Finger-spelling alphabet for ASL . . . . .	19
2.9	Structural design of speech to sign language interpreter . . . . .	20
2.10	Structural design of Spanish to sign language translation . . . . .	22
2.11	<i>eSign</i> signing avatar and <i>ViSiCAST</i> signing avatar . . . . .	24
2.12	CWASA: Marc and Anna . . . . .	25
3.1	Architectural design of the proposed text to sign language interpreter	36
3.2	Client UI interface . . . . .	37
3.3	Invoking Microservices Language Parser Handler via Postman . . . . .	38
3.4	Response of the LanguageParserHandler via Postman . . . . .	39
3.5	Calling the Parser Handler microservice from JavaScript code . . . . .	40
3.6	passed_string collection in mongoDb . . . . .	40
3.7	Connect with MongoDB from Flask . . . . .	41
3.8	Lemmatizing tokens with python’s NLTK . . . . .	42
3.9	Generating Fingerspelling . . . . .	44
3.10	Client Event Handler . . . . .	44

3.11 SiGML Converter . . . . .	46
3.12 SiGML Dictionary . . . . .	46
4.1 Utility tree . . . . .	56

## Nomenclature

ASL American Sign Language

ASR Architectural Significant Requirments

ATAM Architecture Trad-off Analysis method

BSL British Sign Language

HamNoSys Hamburg Notation System

SigML Signing Gesture Markup Language

SL Sign Language

SOV Subject Object Verb

SSL Sinhala Sign Language

SVO Subject Verb Object

XML Extensible Markup Language