

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RÚBIA MEDIANEIRA DENARDI

**AGA-Sign: Animador de Gestos Aplicado à  
Língua de Sinais**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de  
Mestre em Ciência da Computação

Prof. Dr. Paulo Fernando Blauth Menezes  
Orientador

Prof. Dr. Antônio Carlos da Rocha Costa  
Co-orientador

Porto Alegre, novembro de 2006

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Denardi, Rúbia Medianeira

AGA-Sign: Animador de Gestos Aplicado à Língua de Sinais / Rúbia Medianeira Denardi. – Porto Alegre: PPGC da UFRGS, 2006.

82 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2006. Orientador: Paulo Fernando Blauth Menezes; Co-orientador: Antônio Carlos da Rocha Costa.

1. Língua de Sinais. 2. LIBRAS. 3. *SIGNWRITING*. 4. AGA. 5. Animação de sinais. I. Menezes, Paulo Fernando Blauth. II. Costa, Antônio Carlos da Rocha. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof<sup>a</sup>. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Você não sente, não vê  
Mas eu não posso deixar de dizer, Meu Amigo  
Que uma nova mudança em breve vai acontecer  
O que há algum tempo era jovem novo  
Hoje é antigo  
E precisamos todos rejuvenescer...”*

— BELCHIOR

## AGRADECIMENTOS

Agradeço a minha família, que mesmo distante se faz presente na minha vida. Aos meus pais e a Tutty, que sempre apoiam minhas escolhas, muito obrigada!

Agradeço aos professores Paulo Blauth Menezes e Antônio Carlos da Rocha Costa pela orientação dedicada a este trabalho. Agradeço pela paciência e pela calma a mim transmitidas.

Agradecimento especial ao amigo Tiarajú Asmuz Diverio, meu chefe! Obrigada pela amizade e pela confiança.

Agradeço aos amigos Mônica e Rafael pela força de sempre. Ao “Trio” Clarissa, Sanger e Diego Contessa pela simpatia e carinho com que me receberam no LabTeC. Aos amigos que “não valem nada” Rafael Ennes, Lapys, André Martinotto e Righi pelas divertidas conversas nas “cidades baixas” da vida. Ao amigo Rodrigo Machado pelas dicas de “Teoria”, pelas conversas sobre Belchior e sobre a vida! Aos amigos Carlos Hölbig, Kaqui, Marnes, Rodrigo Kassick, Carlos Machado, Juliana Vizzotto, Seonara, Luciano Ramis e Pricila Sleifer. A todos vocês meu muito obrigada!

Agradeço, também, de coração as amigas Renata Z., Mariusa, Graciela, Márcia e Cibele Christ pelo carinho, amizade, atenção, pela força nas horas difíceis e pela preocupação com o meu trabalho, pela alegria e pelas risadas maravilhosas.

Agradeço ao Instituto de Informática (II) e ao Programa de Pós-Graduação em Computação (PPGC) pelos serviços prestados. Agradeço também ao CNPq e a Secretaria de Educação a Distância pelo apoio financeiro aos projetos de que participei durante o mestrado.

# SUMÁRIO

|   |    |
|---|----|
| <b>LISTA DE ABREVIATURAS E SIGLAS</b> . . . . .             | 7  |
| <b>LISTA DE FIGURAS</b> . . . . .                           | 8  |
| <b>LISTA DE TABELAS</b> . . . . .                           | 11 |
| <b>RESUMO</b> . . . . .                                     | 12 |
| <b>ABSTRACT</b> . . . . .                                   | 13 |
| <b>1 INTRODUÇÃO</b> . . . . .                               | 14 |
| <b>2 VISÃO GERAL DO AGA-SIGN</b> . . . . .                  | 17 |
| <b>3 PROCESSAMENTO DE LÍNGUAS DE SINAIS (PLS)</b> . . . . . | 19 |
| <b>3.1 LIBRAS</b> . . . . .                                 | 22 |
| <b>3.2 Estrutura da LIBRAS</b> . . . . .                    | 22 |
| 3.2.1 Sistema Fonológico . . . . .                          | 22 |
| 3.2.2 Sistema Morfológico . . . . .                         | 23 |
| 3.2.3 Sistema Sintático . . . . .                           | 24 |
| 3.2.4 Sistema Semântico . . . . .                           | 25 |
| <b>3.3 Aquisição da Leitura/Escrita</b> . . . . .           | 25 |
| <b>4 SISTEMA DE ESCRITA DE SINAIS</b> . . . . .             | 27 |
| <b>4.1 SignWriting</b> . . . . .                            | 28 |
| 4.1.1 Símbolos SW . . . . .                                 | 28 |
| 4.1.2 Sinal-Símbolo-Sequência e SignSpelling . . . . .      | 29 |
| 4.1.3 SymbolBank e SignSpelling . . . . .                   | 30 |
| 4.1.4 Editores . . . . .                                    | 33 |
| <b>4.2 SWML</b> . . . . .                                   | 35 |
| <b>4.3 Uso do SignWriting no Brasil</b> . . . . .           | 36 |
| <b>5 AGA</b> . . . . .                                      | 38 |
| <b>5.1 Modelo AGA</b> . . . . .                             | 39 |
| 5.1.1 Autômatos Finitos com Saída . . . . .                 | 39 |
| 5.1.2 Função de Saída Contextual . . . . .                  | 40 |
| 5.1.3 Função de Transição Temporal . . . . .                | 40 |
| 5.1.4 Função Descrição . . . . .                            | 41 |
| <b>5.2 Implementação do Modelo AGA</b> . . . . .            | 41 |
| 5.2.1 Linguagem AgaML . . . . .                             | 42 |

|            |  |           |
|------------|--|-----------|
| 5.2.2      | Especificação da Animação . . . . .  | 43        |
| 5.2.3      | Especificação do Ator AGA . . . . .  | 44        |
| 5.2.4      | Especificação da Fita de Entrada . . . . .                                     | 45        |
| 5.2.5      | Especificação das Instâncias do Ator AGA . . . . .                             | 47        |
| <b>5.3</b> | <b>AGA Player . . . . .</b>  | <b>47</b> |
| <b>6</b>   | <b>TRADUTOR SWML/AGAML . . . . .</b>   | <b>50</b> |
| <b>6.1</b> | <b>Estudo de Textos SW . . . . .</b>   | <b>50</b> |
| 6.1.1      | Estudo dos Movimentos . . . . .  | 53        |
| 6.1.2      | Resultados do Estudo de Textos SWML . . . . .                                  | 58        |
| 6.1.3      | Definindo Regras de Transições de Estados para Movimentos Verticais . . . . .  | 60        |
| 6.1.4      | Definindo Regras de Transição de Estados para Movimentos Horizontais . . . . . | 61        |
| 6.1.5      | Definindo Regras de Transição de Estados para Movimentos Circulares . . . . .  | 62        |
| <b>6.2</b> | <b>Tradutor de Textos SWML/AgaML . . . . .</b>                                 | <b>64</b> |
| <b>7</b>   | <b>CONCLUSÃO . . . . .</b>   | <b>71</b> |
| <b>7.1</b> | <b>Trabalhos Futuros . . . . .</b>   | <b>73</b> |
|            | <b>REFERÊNCIAS . . . . .</b>   | <b>75</b> |
|            | <b>ANEXO A EXEMPLOS DE SOLETRAÇÕES . . . . .</b>                               | <b>78</b> |
| <b>A.1</b> | <b>Quando não existe mão dominante . . . . .</b>                               | <b>78</b> |
| <b>A.2</b> | <b>Movimentos dominantes e não-dominantes . . . . .</b>                        | <b>78</b> |
| <b>A.3</b> | <b>Expressões Faciais . . . . .</b>  | <b>79</b> |
| <b>A.4</b> | <b>Guia de Pronúncia . . . . .</b>   | <b>79</b> |
|            | <b>ANEXO B DOCUMENTO SWML . . . . .</b>  | <b>81</b> |

## LISTA DE ABREVIATURAS E SIGLAS

AgaML AGA Markup Language

ASL Língua de Sinais Americana

LIBRAS Língua Brasileira de Sinais

PLN Processamento de Línguas Naturais

PLS Processamento de Línguas de Sinais

SSS Sinal-Símbolo-Sequência

SW *SignWriting*

SWML SignWriting Markup Language

## LISTA DE FIGURAS

|              |  |    |
|--------------|--|----|
| Figura 2.1:  | Aga-Sign . . . . .   | 17 |
| Figura 3.1:  | Exemplos que mostram a mesma configuração de mãos . . . . .  | 23 |
| Figura 3.2:  | Exemplos de sinais sem e com movimento . . . . .   | 23 |
| Figura 3.3:  | Exemplos de sinais com diferentes direções . . . . .   | 23 |
| Figura 3.4:  | Exemplo de sinal próprio que indica o gênero . . . . .   | 24 |
| Figura 3.5:  | Exemplo de sinais próprios que indicam o grau . . . . .  | 24 |
| Figura 3.6:  | Exemplo de sinais que indicam tempo . . . . .  | 24 |
| Figura 3.7:  | Exemplo de datilologia . . . . .   | 25 |
| Figura 3.8:  | Exemplo de presença ou não presença da iconicidade . . . . .   | 25 |
| Figura 4.1:  | Exemplos da Notação de Stokoe . . . . .  | 27 |
| Figura 4.2:  | Exemplos da Notação HamNoSys (GERMAN SIGN LANGUAGE;<br>DEAF, 2004) . . . . .   | 28 |
| Figura 4.3:  | Alguns símbolos do <i>SignWriting</i> . . . . .  | 28 |
| Figura 4.4:  | Editor de Sinais SignBank (DAC, 2004a) . . . . .   | 31 |
| Figura 4.5:  | Divisão de Sílabas em <i>SignWriting</i> (DAC, 2004a) . . . . .  | 31 |
| Figura 4.6:  | Soletração básica (DAC, 2004a) . . . . .   | 32 |
| Figura 4.7:  | Soletração detalhada (DAC, 2004a) . . . . .  | 33 |
| Figura 4.8:  | Sinal “surdo” . . . . .  | 33 |
| Figura 4.9:  | Editor Sign Writer . . . . .   | 34 |
| Figura 4.10: | Editor SWEdit . . . . .  | 35 |
| Figura 4.11: | Intercâmbio de arquivos <i>SW</i> entre diferentes aplicações (COSTA,<br>2006a) . . . . .  | 36 |
| Figura 4.12: | Trecho de um texto <i>SWML</i> . . . . .   | 36 |
| Figura 5.1:  | Atores especificados no modelo AGA . . . . .   | 38 |
| Figura 5.2:  | Troca de imagens do ator mão direita provocadas pela leitura da fita . . . . .   | 39 |
| Figura 5.3:  | Alternativas de construção dos atores AGA . . . . .  | 41 |
| Figura 5.4:  | Estrutura de instânciação dos atores . . . . .   | 43 |
| Figura 5.5:  | Trecho da DTD com os elementos <i>AGA</i> , <i>HEAD</i> , <i>TITLE</i> , <i>AUTHOR</i> e<br><i>SUBJECT</i> . . . . .   | 44 |
| Figura 5.6:  | Trecho da DTD com os elementos <i>ACTOR</i> , <i>OUTPUT</i> , <i>DESCF</i> , <i>DES-</i><br><i>CRPTION</i> , <i>TRANSF</i> , <i>FROM</i> e <i>TO</i> . . . . . | 45 |
| Figura 5.7:  | Exemplo do elemento <i>ACTOR</i> para a especificação do ator <i>mão direita</i> . . . . .   | 46 |
| Figura 5.8:  | Trecho da DTD com os elementos <i>TAPE</i> , <i>CEL</i> e <i>GROUP</i> . . . . .   | 46 |
| Figura 5.9:  | Exemplo do elemento <i>TAPE</i> para especificação da fita de entrada . . . . .  | 46 |
| Figura 5.10: | Trecho da DTD com o elemento <i>INSTANCE</i> e <i>USE</i> . . . . .  | 47 |



|   |    |
|---|----|
| Figura 5.11: Exemplo do elemento INSTANCE para a criação de instâncias . . . . .  | 47 |
| Figura 5.12: Painel de controle do AGA Player visualizado pelo observador . . . . .   | 48 |
| Figura 6.1: Representação de um Contato duplo (COSTA, 2006b) . . . . .  | 54 |
| Figura 6.2: Contato Bater - sinal “Pagar” (COSTA, 2006b) . . . . .  | 54 |
| Figura 6.3: Contato Escovar - sinal “Morrer” (COSTA, 2006b) . . . . .   | 54 |
| Figura 6.4: Contato Esfregar - sinal “Vinho” (COSTA, 2006b) . . . . .   | 55 |
| Figura 6.5: Representação de movimentos verticais (COSTA, 2006b) . . . . .  | 55 |
| Figura 6.6: Movimento vertical - sinal “Exato” (COSTA, 2006b) . . . . .   | 56 |
| Figura 6.7: Representação de movimentos horizontais para trás e para frente<br>(COSTA, 2006b) . . . . .   | 56 |
| Figura 6.8: Movimento horizontal - sinal <i>Eles</i> (COSTA, 2006b) . . . . .   | 57 |
| Figura 6.9: Movimentos Circulares para cima e para baixo (COSTA, 2006b) . . . . .   | 57 |
| Figura 6.10: Exemplo de movimento circular, sinal " <i>Computador</i> " . . . . .   | 58 |
| Figura 6.11: Movimentos circulares para frente e para trás e movimentos circula-<br>res para cima e para frente e para baixo e para trás (COSTA, 2006b) . . . . . | 58 |
| Figura 6.12: Exemplo de movimento circular para trás, sinal " <i>Sozinho</i> " . . . . .  | 58 |
| Figura 6.13: Exemplo de movimento circular para trás, sinal " <i>Brincar</i> " . . . . .  | 58 |
| Figura 6.14: Exemplo de um trecho de texto analisado . . . . .  | 59 |
| Figura 6.15: Exemplo de um gesto escrito em SW com movimento vertical . . . . .   | 59 |
| Figura 6.16: Exemplo de um gesto escrito em SW com movimentos combinados . . . . .  | 59 |
| Figura 6.17: Direções dos movimentos verticais . . . . .  | 60 |
| Figura 6.18: Escrita do Sinal <i>Chorar</i> . . . . .   | 60 |
| Figura 6.19: Exemplo de deslocamento para o sinal <i>Chorar</i> . . . . .   | 60 |
| Figura 6.20: Exemplo algorítmico para regras de movimentos verticais . . . . .  | 61 |
| Figura 6.21: Direções dos movimentos horizontais . . . . .  | 61 |
| Figura 6.22: Escrita do Sinal <i>Idéia</i> . . . . .  | 61 |
| Figura 6.23: Exemplo do deslocamento do Sinal <i>Idéia</i> . . . . .  | 62 |
| Figura 6.24: Exemplo algorítmico para regras de movimentos horizontais . . . . .  | 62 |
| Figura 6.25: Escrita do Sinal " <i>Importante</i> " . . . . .   | 62 |
| Figura 6.26: Deslocamentos das mãos para o Sinal " <i>Importante</i> " . . . . .  | 63 |
| Figura 6.27: Exemplo algorítmico para regras para movimentos circulares para<br>cima e para baixo . . . . .   | 63 |
| Figura 6.28: Escrita do Sinal " <i>Sozinho</i> " . . . . .  | 63 |
| Figura 6.29: Deslocamento das mãos para o sinal " <i>Sozinho</i> " . . . . .  | 64 |
| Figura 6.30: Exemplo algorítmico para regras para movimentos circulares para<br>frente e para trás . . . . .  | 64 |
| Figura 6.31: Entrada e saída de dados do Tradutor . . . . .   | 64 |
| Figura 6.32: SSS de cada símbolo como entrada para o Tradutor . . . . .   | 65 |
| Figura 6.33: Processamento das informações . . . . .  | 65 |
| Figura 6.34: Dados de saída do Tradutor - documento AgaML . . . . .   | 66 |
| Figura 6.35: Sinal <i>Idéia</i> . . . . .   | 66 |
| Figura 6.36: Processamento dos SSS's do sinal <i>Idéia</i> . . . . .  | 66 |
| Figura 6.37: Resultado do processamento dos SSS's do sinal <i>Idéia</i> . . . . .   | 67 |
| Figura 6.38: Elemento ACTOR para a especificação do ator <i>cabeça</i> . . . . .  | 67 |
| Figura 6.39: Elemento ACTOR para a especificação do ator <i>mão direita</i> . . . . .   | 67 |
| Figura 6.40: Trecho do SWML com o SSS do símbolo da <i>cabeça</i> . . . . .   | 68 |
| Figura 6.41: SSS do símbolo da <i>cabeça</i> . . . . .  | 68 |

|  |    |
|--|----|
| Figura 6.42: Elemento TAPE para a especificação de fita de entrada da <i>mão direita</i>               | 68 |
| Figura 6.43: Elemento INSTANCE para instanciar os atores e suas respectivas fitas de entrada . . . . . | 68 |
| Figura 6.44: Exemplo de applet do sinal <i>Idéia</i> . . . . .   | 69 |
| Figura 6.45: AGA Player da animação do sinal <i>Idéia</i> . . . . .                                    | 69 |
| Figura 6.46: AGA Player da animação do sinal <i>Idéia</i> . . . . .                                    | 69 |

## LISTA DE TABELAS

|              |  |    |
|--------------|--|----|
| Tabela 4.1:  | Categorias <i>SW</i> . . . . .                     | 29 |
| Tabela 4.2:  | SSS de um símbolo <i>SW</i> . . . . .              | 30 |
| Tabela 6.1:  | SSS de um símbolo <i>SW</i> . . . . .              | 50 |
| Tabela 6.2:  | Categoria 1 - Mãos . . . . .                       | 51 |
| Tabela 6.3:  | Categoria 2 - Movimentos . . . . .                 | 51 |
| Tabela 6.4:  | Categoria 3 - Expressões Faciais . . . . .         | 52 |
| Tabela 6.5:  | Categoria 4 - Cabeça . . . . .                     | 52 |
| Tabela 6.6:  | Categoria 5 - Tronco . . . . .                     | 52 |
| Tabela 6.7:  | Categoria 6 - Membro . . . . .                     | 53 |
| Tabela 6.8:  | Categoria 7 - Localização . . . . .                | 53 |
| Tabela 6.9:  | Categoria 8 - Dinâmicas e Pontuação . . . . .      | 53 |
| Tabela 6.10: | SSS dos movimentos de contato analisados . . . . . | 54 |
| Tabela 6.11: | SSS do Movimento Vertical . . . . .                | 55 |
| Tabela 6.12: | SSS do Movimento Horizontal . . . . .              | 56 |
| Tabela 6.13: | SSS dos Movimentos Circulares . . . . .            | 57 |

## RESUMO

A expansão da Internet e o crescente desenvolvimento de tecnologias para a *Web* fazem com que um grande número de pessoas com necessidades distintas procurem nelas as informações de que necessitam, utilizando a Internet como um meio de ensino e aprendizagem. Motivado por isso, procura-se atender a comunidade surda com a obtenção de um Animador de Gestos aplicado à Língua de Sinais, o AGA-Sign, disponível em ambiente *Web*, com o objetivo de auxiliar na prática da escrita de sinais e na familiarização com a língua.

Este trabalho apresenta uma aplicação para geração automatizada de animações de gestos aplicado à Língua de Sinais a partir de textos escritos em *SignWriting* - sistema de escrita para Língua de Sinais que dispõe de expressões gráficas para descrever os movimentos das mãos, dos braços, além de expressões faciais, fazendo com que o sistema possa representar qualquer língua de sinais. Os sinais usados para o desenvolvimento da aplicação foram elaborados a partir da LIBRAS por ser a língua oficial dos surdos brasileiros e as animações foram geradas através do modelo AGA (animação gráfica baseada na Teoria dos Autômatos).

**Palavras-chave:** Língua de Sinais, LIBRAS, *SignWriting*, AGA, animação de sinais.

## AGA-Sign: Animator of Gestures applied to the Sign Languages

### ABSTRACT

The Internet expansion and the increasing development of *Web* technologies make a great audience with distinct necessities search the information they need, using the Internet as a mean for teaching and learning. In this context, this work tries to support the deaf community by developing a Gesture Animator applied to the Sign Language, the AGA-Sign, available in the *Web* environment, which goal is assisting the writing practice of signs and the familiarization with the language.

This work presents an application for automatic generation of gestures animations applied to the Sign Language from texts written in *SignWriting* - a Sign Language writing system that uses graphical expressions to describe hands and arms movements, beyond face expressions, which makes the system capable of representing any sign language. The signs used for the development of the application were elaborated based in LIBRAS, that is the official brazilian deaf language and the animations were generated through AGA model (graphical animation based in the Automata Theory).

**Keywords:** Sign Languages, LIBRAS, *SignWriting*, animator based on Automata Theory, sign animation.

# 1 INTRODUÇÃO

A expansão da Internet e o crescente desenvolvimento de tecnologias para a *Web* fazem com que um grande número de pessoas com necessidades distintas procurem nelas as informações que desejam, utilizando a Internet como um meio de ensino, aprendizagem e inclusão social.

Devido à diversidade de pessoas que utilizam a Internet com esses fins, existem várias pesquisas que buscam cada vez mais desenvolver tecnologias, metodologias e ferramentas que atendam às necessidades destas pessoas. Para o caso de portadores de necessidades especiais, particularmente a comunidade surda (a qual se dedica este trabalho), encontram-se algumas pesquisas como as apresentadas em (COSTA, 2006c; FIAP, 2004; INSTITUTE, 2004).

Levando-se em conta o número significativo de portadores de necessidades especiais, que segundo o Censo de 2000 do Instituto Brasileiro de Geografia e Estatística - IBGE, são 5.685.956 portadores de necessidades educativas especiais - PNNE, com dificuldades de audição no Brasil, em 2002 o Congresso Nacional e a Presidência da República aprovaram e sancionaram a Lei n.º 10.436, de 24 de abril de 2002, que reconhece a LIBRAS, Língua Brasileira de Sinais, como meio legal de comunicação e expressão da comunidade surda, e que estimula as escolas à sua adoção como recurso educacional.

Um dos maiores problemas na educação do surdo provém do ensino e aprendizagem da língua portuguesa, pois, até bem pouco tempo, o ensino do português ao aluno surdo se encaminhava da mesma forma que ao aluno ouvinte. A consequência mais grave era que se mantinha a idéia de que o surdo era cognitivamente comprometido em relação aos ouvintes. Entretanto, quando o ensino de português para o surdo passa a ser visto sob outro ponto de vista, como segunda língua, a situação muda de perfil.

Segundo (STUMPF, 2000), a surdez é uma diferença que faz de seus portadores pessoas que passam a se comunicar de forma essencialmente visual, portanto de forma perfeitamente compatível com a utilização do computador como ferramenta educacional, fazendo com que esse equipamento e a tecnologia de processamento de informação que o acompanha possam se tornar instrumentos de uso prioritário na educação de surdos. Partindo deste pensamento, é importante que se descubra a riqueza linguística das línguas de sinais (especialmente a LIBRAS) e a oportunidade de pesquisa e desenvolvimento de aplicações de Processamento de Língua de Sinais (PLS) que podem ser extremamente úteis do ponto de vista social e educacional, para os surdos (COSTA, 2006d).

Instituições como a USP, UCPel e PUC-RS desenvolvem estudos (CAPOVILLA (2006; 2006; 2004; 2003)), dicionários e ferramentas que facilitam o processo de ensino e aprendizagem da comunidade surda, como o Dicionário de LIBRAS (CAPOVILLA et al., 1999), Editor de sinais SW-Edit (TORCHELSEN; COSTA; DIMURO, 2002) e o SIGNSIM (CAMPOS; GIRAFFA; SANTAROSA, 2000).

A proposta deste trabalho é obter um animador de gestos aplicado à Língua de Sinais que rode em ambiente Web, o AGA-Sign. o animador consiste em uma aplicação responsável pela animação de gestos <sup>1</sup> (ou sinais) voltados à Língua de Sinais e que pode ser usada tanto por pessoas surdas como por pessoas ouvintes. Esta aplicação possui construções especialmente projetadas para:

- Ensino/aprendizagem de deficientes auditivos;
- Ensino/aprendizagem de deficientes auditivos x ouvintes;
- Comunicação entre deficientes auditivos;
- Comunicação entre deficientes auditivos x ouvintes.

O AGA-Sign está inserido no contexto de dois projetos de pesquisa: SignNet/PLN (COSTA, 2006c) e Hyper Seed (MENEZES, 2004), pois faz uso de ferramentas usadas e desenvolvidas por esses projetos.

O Projeto SignNet/PLN (Processamento de Línguas de Sinais - Língua Brasileira de Sinais) tem como meta a construção de ferramentas de processamento de textos de línguas de sinais, baseadas no sistema de escrita *SignWriting* (SUTTON, 2005) e em formatos XML de armazenamento para demonstração da viabilidade e utilidade da transposição, para as línguas de sinais, das técnicas consolidadas da lingüística computacional, desenvolvidas originalmente para as línguas orais. O fundamento pedagógico da proposta consiste na proposição de que, se é verdade que as línguas de sinais são as línguas naturais dos surdos, então a alfabetização dos mesmos deve ser feita preferencialmente nessas línguas e não nas línguas orais das sociedades em que eles vivem. No caso do Brasil, isso significa que a alfabetização dos surdos deve ser feita na LIBRAS (Língua Brasileiro de Sinais).

O Projeto Hyper Seed tem como proposta buscar uma fundamentação matemática para a unificação de especificações (baseadas na Teoria dos Autômatos e Teoria das Categorias) de hiperdocumentos e animações, e, baseado nesta fundamentação, desenvolver um protótipo para suporte ao desenvolvimento de conteúdo hipermídia/multimídia que permita uma integração natural a ambientes voltados para o EAD via WWW. Um dos objetivos alcançados pelo projeto foi o desenvolvimento de um modelo baseado na Teoria dos Autômatos para criação de animações em ambiente *Web*, o AGA (ACCORSI, 2002), usado neste trabalho para fazer a animação de sinais.

A animação de sinais em ambiente *Web*, gerada a partir de textos de línguas de sinais escritos em *SignWriting*, vem sendo proposta há alguns anos e está inserida no contexto da área de Processamento de Línguas Naturais (PLN), que estuda os sistemas computacionais para compreensão e geração de línguas naturais faladas e escritas.

O uso do AGA se justifica por questões de projeto e por apresentar características que favorecem sua aplicação na especificação e controle de animações para a *Web*, como a redução de espaço de armazenamento e o encapsulamento das propriedades estéticas e comportamentais dos atores em uma unidade básica, o ator AGA, favorecendo seu reuso em diferentes animações. Tem-se como uns dos objetivos específicos testar a aplicabilidade do AGA.

O AGA-Sign é uma aplicação que reúne ferramentas já existentes, como o sistema de escrita *SignWriting*, a SWML (COSTA, 2006a) e o AGA. O Capítulo 2 contém uma

---

<sup>1</sup>Não se restringe à animação de sinais. O usuário pode produzir gestos que não constituem um sinal.

breve apresentação do AGA-Sign, com as ferramentas que fazem parte do modelo e suas características.

Para contextualizar o trabalho, o Capítulo 3 apresenta as características do Processamento de Línguas Naturais e das Línguas de Sinais, especialmente a LIBRAS, por ser a língua materna dos surdos brasileiros e utilizada neste trabalho.

No Capítulo 4 é descrito o sistema de escrita *SignWriting*, seus componentes e algumas ferramentas que fazem uso de seu conjunto de símbolos, como os editores de sinais, e a linguagem SWML, que é responsável pelo armazenamento, troca e processamento de textos SW. Estudo indispensável para o desenvolvimento do trabalho, já que o sistema está baseado em textos escritos em *SignWriting* e convertidos para SWML.

O Capítulo 5 apresenta o modelo de animação AGA e sua linguagem de especificação, a AgaML.

Para contemplar o objetivo geral, um dos objetivos específicos é identificar as características fonológicas e morfológicas dos sinais escritos em *SignWriting* e determinar as regras de interpretação dos símbolos constituintes dos sinais escritos, principalmente dos símbolos de movimento, para uma possível especificação dos processos de animação correspondente a essas características. O estudo dos símbolos do *SignWriting* é apresentado no Capítulo 6.

Outro objetivo específico é adaptar a especificação da animação dos atores, fitas e instâncias dos atores AGA à especificação de animações de sinais. Para isso, foi desenvolvido um Tradutor de Textos SWML/AgaML, responsável pela geração de um documento com as informações sobre cada símbolo que compõe o sinal e de como deve se comportar durante a animação, apresentado também no Capítulo 6.

Por fim, o Capítulo 7 apresenta as conclusões e os trabalhos futuros.



## 2 VISÃO GERAL DO AGA-SIGN

Neste capítulo será apresentada uma visão geral do AGA-Sign, uma aplicação voltada para a animação de gestos aplicados à Língua de Sinais.

Para atingir-se o objetivo principal do trabalho, que é a animação de gestos (ou sinais), foram seguidas algumas etapas (Figura 2.1), fazendo-se uso de algumas ferramentas e sistemas relacionados à Língua de Sinais e uma ferramenta relacionada a animações.

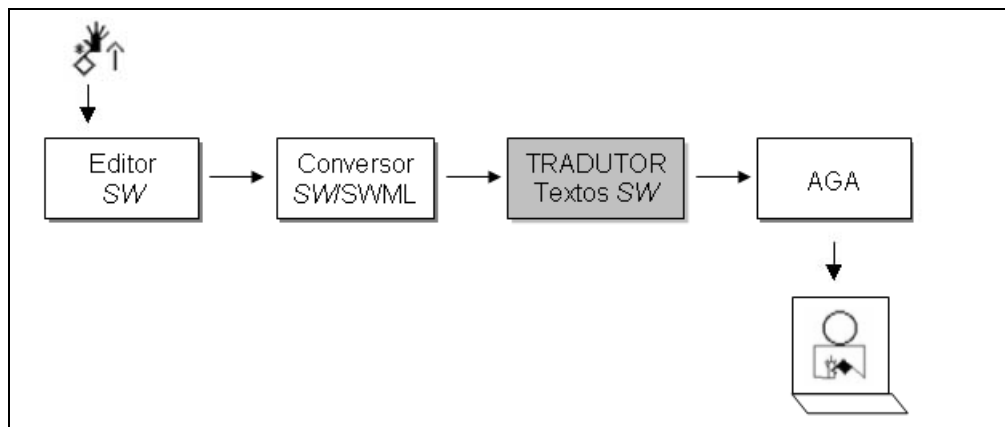


Figura 2.1: Aga-Sign

As ferramentas que fazem parte da aplicação são listadas a seguir e serão apresentadas com detalhes nos capítulos seguintes.

- **Editor SW:** Esta primeira ferramenta é usada para a representação de sinais, ou gestos, que são escritos a partir de um sistema de escrita de sinais. Usa-se no trabalho o sistema *SignWriting(SW)*. Essas representações são elaboradas em editores que dispõem dos símbolos do SW e todos os sinais são sintetizados a partir da LIBRAS;
- **Conversor SW/SWML:** Após a edição dos sinais, estes arquivos de sinais são convertidos para textos SWML, gerados pelo Conversor SW/SWML (COSTA, 2006a). SWML é a linguagem criada para suportar adequadamente o armazenamento, troca e processamento de textos do SW. Um texto convertido em SWML apresenta as características dos símbolos pertencentes ao sinal, como por exemplo a posição do símbolo e seu formato.
- **Tradutor de Textos SW:** é a ferramenta de ligação entre um texto SWML que representa um sinal e a animação deste sinal. Esta tradução é feita a partir da aplicação de algumas regras criadas através do estudo dos símbolos SW. A tradução possibilita a

especificação da animação de sinais, gerando um documento AgaML, vocabulário XML desenvolvido especialmente para especificar as animações no modelo AGA.

- AGA: é um modelo de animação que especifica a animação a partir de um conjunto de atores (objetos) e suas respectivas variações durante a animação. Visto as adaptações ocorridas na especificação dos atores AGA, fitas e instâncias, dá-se ao animador o nome de AGA-Sign, um animador de gestos aplicados a Linguagem de Sinais.

### 3 PROCESSAMENTO DE LÍNGUAS DE SINAIS (PLS)

Este capítulo apresenta a área em que o trabalho está inserido: Processamento de Línguas Naturais (PLN), uma sub-área da Inteligência Artificial, que estuda os sistemas computacionais para compreensão e geração de línguas naturais faladas e escritas.

O PLN apresenta-se como um campo de estudos bastante heterogêneo e fragmentado, acumulando uma vasta literatura e agregando pesquisadores das mais variadas especialidades, com formação acadêmica, embasamento teórico e interesses também bastante diversos. Estudar o PLN é fornecer subsídios para a implementação de programas computacionais construídos para o fim específico de manipulação de objetos lingüísticos.

Segundo MARTINS et. al (1999), no uso cotidiano da língua, a comunicação entre os falantes se dá através de textos, embora esses mesmos falantes possuam uma consciência intuitiva das unidades mínimas da língua. Em termos de análise lingüística, pode-se dizer que o texto é a unidade maior na estrutura de uma língua natural, pois reúne em si informações de diversas naturezas que, por sua vez constitui no objeto de estudo de alguns campos específicos na área da Lingüística. A tarefa do lingüista, é identificar e compreender esses segmentos lingüísticos e, a partir daí, apresentar uma descrição do comportamento desses elementos na realização da linguagem verbal. Nesse processo de descrição, então, costuma-se privilegiar os segmentos menores isolados do texto, tornando esse mesmo texto objeto de estudo particular de uma área da Lingüística.

Em PLN o material de entrada do processamento são textos que devem ser analisados, ou seja, recortados em unidades menores para a compreensão completa dos mecanismos de operação envolvidos em cada uma dessas unidades. Os tipos de informações lingüísticas que são manipuladas pelo computador no processamento automático da língua são as seguintes:

- **Estrutura Linguística.** Um processador lingüístico costuma recortar o texto em segmentos denominados sentenças (S). A análise lingüística automática que opera nesse nível é conhecida como análise sentencial, isto é, o tratamento de um texto é promovido de sentença a sentença, sendo ela, portanto, a primeira unidade menor do processamento. Uma sentença pode ser definida como a unidade mínima da comunicação, uma vez que se apresenta como um enunciado dotado de expressão completa de sentido. Ela também é conhecida através das denominações de frase e oração, comumente diferenciadas na Gramática Tradicional da Língua Portuguesa. No âmbito do PLN, porém, fala-se em sentença para se dirigir aos segmentos organizados das seguintes formas:

1. sentenças constituídas de uma palavra. Ex.: Atenção!

2. sentenças constituídas de um conjunto de palavras no qual se verifica a presença de um verbo (ou locução verbal), ainda que esse verbo esteja oculto. Ex.: (a) A moça toca piano muito bem. [presença de verbo]; (b) O jogo tinha terminado. [presença de locução verbal]
  3. sentenças constituídas de algumas palavras dentre as quais não há verbo. Ex.: Que falsa modéstia, meu Deus!
- **Níveis de Processamento.** As palavras podem ser caracterizadas de várias maneiras, dependendo das regras da descrição lingüística. Isso é devido ao fato de que as palavras possuem propriedades de natureza distinta, refletindo o comportamento que elas adquirem quando combinadas entre si na atividade comunicativa. Dessa forma, podemos definir uma palavra pelas regras:
    - *fonética-fonológica*: quando se trata de compreender a identidade sonora dos elementos que constituem a palavra;
    - *morfológica*: quando as unidades mínimas dotadas de significado são isoladas para a compreensão do processo de formação e flexão das palavras.
    - *sintática*: quando a distribuição das palavras resulta em determinadas funções que elas desempenham na sentença.
    - *semântica*: quando o conteúdo significativo da palavra implica relações de natureza ontológica e referencial para a identificação dos objetos no mundo.
    - *pragmática-discursiva*: quando a força expressiva das palavras remete à identificação dos objetos do mundo em termos do seu contexto de enunciação e condições de produção discursiva.

- **Informações Linguísticas relativas as regras:**

1. Fonético-fonológicas: A análise lingüística denomina de Fonologia o estudo do efeito acústico das formas sonoras da língua. Já a Fonética ocupa-se da descrição dos sons da fala e das condições pelas quais esses sons são reconhecidos e produzidos pelos falantes de uma língua.
2. Morfológicas: As palavras da língua também podem ser segmentadas em termos do seu conteúdo significativo. As unidades mínimas dotadas de significado (gramatical ou lexical) são denominadas morfemas e se constituem no objeto de estudo da Morfologia.
3. Sintáticas: A organização das palavras na sentença acarreta a definição desses itens lexicais em termos de suas funções gramaticais. Trata-se de reconhecer as regras pelas quais a distribuição das formas são determinadas e esse exercício é o objeto de estudo da Sintaxe.
4. Semânticas: As relações envolvidas no plano do significado das palavras em busca de alcançarem certo sentido no escopo da sentença é a matéria de investigação da Semântica. O significado é inerente ao signo lingüístico e está presente não só na palavra como uma unidade completa, mas nas suas unidades constitutivas. Da mesma forma, fala-se em significado de expressões, de sentenças, enfim, de unidades mais complexas da língua. Grande parte do esforço do tratamento semântico em PLN deve envolver, então, a compreensão das propriedades semânticas dos itens lexicais para a construção de sentenças semanticamente bem formadas da língua.

5. Pragmático-discursivas: Nesse nível de análise lingüística estão em foco as questões do mundo extralingüístico. Essa noção é amparada pelo fato de que para além das formas e das estruturas, a língua recupera da situação comunicativa diversos fatores que implicam a determinação de certa compreensão das palavras e sentenças. Todo texto é produzido por certos interlocutores, em um tempo e um lugar determinado, o que significa dizer que nenhum texto existe independente dos indivíduos envolvidos na atividade comunicativa e nenhum texto existe sem uma situação de contexto. Quando se examina uma construção lingüística procurando essas relações presentes no ato da fala, na verdade procura-se estudar aquilo que é objeto da Pragmática.

Há anos diversas pesquisas vem sendo desenvolvidas no intuito de criar sistemas que tornem máquinas capazes de comunicar-se através da linguagem falada. Tais esforços incluem desde o processo de reconhecimento e interpretação da linguagem até a síntese de voz que se constitui na geração automática por computador das formas de onda da voz.

O reconhecimento da fala natural envolve várias palavras que tenham um sentido semântico. A fala reconhecida é convertida em texto. O tipo de aplicação mais comum para esta tecnologia é o ditado de documentos, para uso em processadores, planilhas de dados, escrita de e-mails, etc. A síntese da voz é o processo contrário ao do reconhecimento da fala. O sintetizador recebe um texto na forma digital e o transforma em onda sonoras, ou em outras palavras, fazendo uma leitura em voz alta. Um programa de síntese de voz é útil nas situações em que o usuário não pode desviar a atenção para ler algo ou não tem acesso ao texto escrito, seja porque a informação está distante ou porque tenha alguma deficiência visual (DUTOIT, 1997).

Já este trabalho, que envolve o estudo de Línguas de Sinais, sistema de escrita de sinais, ferramentas para suporte ao Processamento de Línguas de Sinais e por fim a animação de sinais através de textos está voltado para os surdos, que utilizam línguas de modalidade gestual-visual em sua comunicação diária. Esses textos são escritos através de notações gráficas que representam as partes do corpo envolvidas nos sinais, assim a animação é perfeitamente compreendida pelo surdo, já que sua língua natural é gestual-visual. Isto é, a animação do sinal está para os surdos assim como a síntese da voz está para os ouvintes.

Os termos oral-auditivo e gestual-visual são diferenciados pela maneira como as línguas são produzidas e percebidas. Para as línguas orais utiliza-se o termo oral-auditivo e para as línguas de sinais, o gestual-visual onde gestual significa o conjunto de elementos lingüísticos manuais, corporais e faciais necessários para a articulação do sinal. Karnopp, em (CAMPOS; COSTA; STUMPF, 1996), cita que nas línguas de sinais, enquanto o emissor constrói uma sentença a partir desses elementos, o receptor utiliza os olhos ao invés dos ouvidos para entender o que está sendo comunicado. Desta forma, já que a informação lingüística é recebida pelos olhos, os sinais são construídos de acordo com as possibilidades perceptuais do sistema visual humano.

Foi a partir do ano de 1960 que William Stokoe, lingüista americano, iniciou um estudo sobre Línguas de Sinais e constatou que a mesma satisfazia todos os critérios de uma língua genuína. Stokoe analisou a estrutura da língua de forma a entender sua organização e as partes dessa estrutura. Como resultado, ele observou em relação à Língua de Sinais Americana que os sinais não eram meras figuras e sim símbolos abstratos e complexos com uma estrutura interna complexa (BATTISON, 1980). Ao contrário do que muitos imaginam, as Línguas de Sinais não são simplesmente mímicas e gestos soltos utilizados pelos surdos para facilitar a comunicação. São línguas com estruturas gramaticais

próprias. Atribui-se às Línguas de Sinais o *status* de língua porque elas também são compostas pelos níveis lingüísticos: o fonológico, o morfológico, o sintático e o semântico. O que é denominado de palavra ou item lexical nas línguas oral-auditivas são denominados sinais nas línguas de sinais (AJA, 2003; FENEIS, 2005).

As línguas de sinais não são universais, cada país possui a sua e pode variar de estado para estado, região para região e também existe a questão dos grupos específicos com suas gírias e jargões.

Para este trabalho, a geração das animações foi feita a partir da Língua Brasileira de Sinais, a LIBRAS, por se tratar da língua oficial dos surdos brasileiros, que foi desenvolvida a partir da língua de sinais francesa.

### 3.1 LIBRAS

A LIBRAS possui estrutura gramatical própria. Os sinais são formados por meio da combinação de formas e de movimentos das mãos e de pontos de referência no corpo ou no espaço. Segundo a legislação vigente, LIBRAS constitui um sistema lingüístico de transmissão de idéias e fatos, oriundos de comunidades de pessoas com deficiência auditiva do Brasil, na qual há uma forma de comunicação e expressão, de natureza visual-motora, com estrutura gramatical própria.

O que diferencia as Línguas de Sinais das demais línguas é a sua modalidade visual-espacial. Assim, uma pessoa que entra em contato com uma Língua de Sinais irá aprender uma outra língua, como o Francês, Inglês etc.

Decretada e sancionada em 24 de abril de 2002, a Lei N° 10.436, no seu artigo 4º, dispõe o seguinte: "O sistema educacional federal e sistemas educacionais estaduais, municipais e do Distrito Federal devem garantir a inclusão nos cursos de formação de Educação Especial, de Fonoaudiologia e de Magistério, em seus níveis médio e superior, do ensino da Língua Brasileira de Sinais - LIBRAS, como parte integrante dos Parâmetros Curriculares Nacionais - PCNs, conforme legislação vigente".

### 3.2 Estrutura da LIBRAS

Segundo Valentini, citado em (CAMPOS; COSTA; STUMPF, 1996), quanto a estrutura da língua, assim como as orais são definidas pelos sistemas fonológico, morfológico e sintático, as línguas de sinais, a partir de sua modalidade gestual-visual, possuem características próprias que também constituem os sistemas acima mencionados.

#### 3.2.1 Sistema Fonológico

Os sinais são formados a partir da combinação da forma e do movimento das mãos e do ponto no corpo ou no espaço onde esses sinais são feitos. Nas línguas de sinais podem ser encontrados os seguintes parâmetros que formarão os sinais:

- Configuração das mãos: São formas das mãos que podem ser da datilologia (alfabeto manual) ou outras formas feitas pela mão dominante (mão direita para os destros ou esquerda para os canhotos), ou pelas duas mãos. Os sinais DESCULPAR e IDADE (Figura 3.1), por exemplo, possuem a mesma configuração de mão (com a letra y). A diferença é que cada uma é produzida em um ponto diferente no corpo.

No sinal DESCULPAR os dedos tocam levemente o queixo. Enquanto que no sinal IDADE o dedo mínimo esfrega o peito para cima por duas vezes.

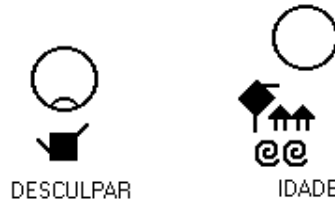


Figura 3.1: Exemplos que mostram a mesma configuração de mãos

- Ponto de articulação: é o lugar onde incide a mão predominante configurada, ou seja, local onde é feito o sinal, podendo tocar alguma parte do corpo ou estar em um espaço neutro.
- Movimento: Os sinais podem ter um movimento ou não. Por exemplo, os sinais PENSAR e EM-PÉ não têm movimento; já os sinais EVITAR e TRABALHAR possuem movimento (Figura 3.2).

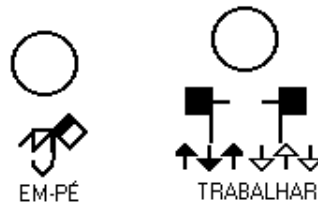


Figura 3.2: Exemplos de sinais sem e com movimento

- Expressão facial e/ou corporal: As expressões faciais / corporais são de fundamental importância para o entendimento real do sinal, sendo que a entonação em Língua de Sinais é feita pela expressão facial.
- Orientação/Direção: Os sinais têm uma direção com relação aos parâmetros acima. Assim, os verbos IR e VIR se opõem em relação à direcionalidade (Figura 3.3).

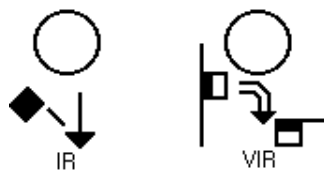


Figura 3.3: Exemplos de sinais com diferentes direções

### 3.2.2 Sistema Morfológico

Assim como as línguas orais possuem um sistema de formação de palavras, as línguas de sinais também possuem. A seguir algumas características quanto ao gênero, grau, tempo, negação e datilologia.

- Gênero: a indicação do sexo é feito colocando-se o sinal de mulher ou de homem independente de serem pessoas ou animais. Outras vezes, é feita com sinais próprios (mãe, pai)(Figura 3.4).



Figura 3.4: Exemplo de sinal próprio que indica o gênero

- Grau: o sinal pode ser diferenciado pela intensidade, movimento, velocidade (p. ex., brabo, muito brabo), ou sinal próprio (p. ex., muito, pouco; grande, pequeno), como no exemplo da Figura 3.5. Bonito, bonitinho, bonitão diferenciam-se somente pela expressão facial.

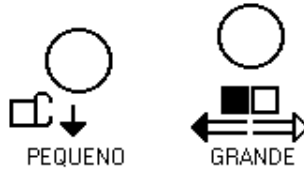


Figura 3.5: Exemplo de sinais próprios que indicam o grau

- Tempo: o passado é indicado por um movimento sobre o ombro até atingir o espaço atrás do ouvido (p. ex.,: ontem) e o passado distante é obtido por um movimento amplo que se estende além das costas (p. ex., há muito tempo) diferenciando-se pela intensidade ou pelo movimento (p. ex., ano, ano passado)(Figura 3.6).

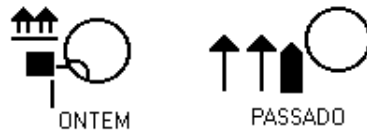


Figura 3.6: Exemplo de sinais que indicam tempo

- Negação: pode-se usar o não (negação feita com a cabeça ou com a mão) mais o sinal ou o sinal próprio. Exemplo: sabe / não sabe (negação feita com o dedo ou o sinal próprio); precisa / não precisa (negação feita com a cabeça); pode / não pode (sinal próprio); acredita / não acredita (negação feita com a cabeça).
- A datilologia (alfabeto manual): usada para expressar nomes de pessoas, lugares e outras palavras que não possuem sinal, estará representada pelas palavras separadas por hífen. Ex.: M-A-R-I-A (Figura 3.8), H-I-P-Ó-T-E-S-E.

### 3.2.3 Sistema Sintático

Segundo Valentini, citado em (CAMPOS; COSTA; STUMPF, 1996), o estudo da descrição quanto à relação dos elementos estruturais e das regras que regem a combinação de sentenças ainda não é completo na LIBRAS. Porém, apresenta regras próprias e básicas.

- Os verbos: são apresentados no infinitivo. Todas as concordâncias e conjugações são feitas no espaço. Ex.: EU QUERER CURSO.





Figura 3.7: Exemplo de datilologia

- As frases: obedecem à estrutura da LIBRAS, e não à do Português. Ex.: VOCÊ GOSTAR CURSO? (Você gosta do curso?)
- Os pronomes pessoais: são representados pelo sistema de apontação. Apontar em LIBRAS é culturalmente e gramaticalmente aceito.

O que se verifica é que as línguas de sinais apresentam pouco uso de preposições e conjunções. Também, por vezes, ocorre omissão dos verbos ser e estar.

Para conversar em LIBRAS não basta apenas conhecer os sinais de forma solta, é necessário conhecer a sua estrutura gramatical, combinando-os em frases.

### 3.2.4 Sistema Semântico

Como ocorre com línguas distintas, o significado de um sinal pode não corresponder exatamente ao de uma palavra equivalente em Português. Palavras com vários significados (polissêmicas), em LIBRAS, podem não ter equivalentes em Português e vice-versa.

Em geral, as línguas de sinais são constituídas de formas que podem ser associadas aos objetos a que se referem, o que se denomina iconicidade. Entretanto, a iconicidade varia de língua para língua o que evidencia a convencionalidade destas formas lingüísticas. Exemplo: o sinal equivalente à "casa" parece com um telhado.



Figura 3.8: Exemplo de presença ou não presença da iconicidade

## 3.3 Aquisição da Leitura/Escrita

Quanto a aquisição da leitura/escrita da língua, pesquisas com filhos surdos de pais surdos estabelecem que a aquisição precoce da Língua de Sinais dentro do lar é um benefício e que esta contribui para o aprendizado da língua oral como segunda língua para os surdos. Justificam este fato salientando que pais surdos sinalizam, utilizam uma linguagem adequada para se comunicarem permitindo, assim, que seus filhos aprendam sua língua na interação (FENEIS, 2005).

Ainda segundo o FENEIS:

”Os estudos em indivíduos surdos demonstram que a Língua de Sinais apresenta uma organização neural semelhante à língua oral, ou seja, que esta se organiza no cérebro da mesma maneira que as línguas faladas. A Língua de Sinais apresenta, por ser uma língua, um período crítico precoce para sua aquisição, considerando-se que a forma de comunicação natural é aquela para

o qual o sujeito está mais bem preparado, levando-se em conta a noção de conforto estabelecido diante de qualquer tipo de aquisição na tenra idade.”

A questão da Língua Portuguesa, em relação a oralização é um processo muito difícil e muito lento. A partir da década de 90 observou-se que a LIBRAS é um processo facilitador não só da comunicação, mas também da veiculação da informação (BESSA, 2004).

## 4 SISTEMA DE ESCRITA DE SINAIS

O sistema de escrita para línguas de sinais adotado neste trabalho é o *SignWriting* (SUTTON, 2005), mas existem outros como o sistema de Stokoe (1965) e o HamNoSys (HANKE, 2004).

William Stokoe desenvolveu um sistema de escrita de sinais a partir do estudo linguístico das línguas de sinais. Foi o primeiro sistema efetivo de escrita de línguas de sinais (Figura 4.1). Stokoe representava as partes componentes da ASL usando símbolos, podia demonstrar como estas partes se juntam para dar forma a uma estrutura linguística idêntica àquela da língua falada. A notação original consistiu em 55 símbolos divididos em três grupos: posição, formato da mão e movimento. Os símbolos foram escritos numa ordem estrita, dependente do significado dentro da palavra. A posição e símbolos de movimento eram icônicos, enquanto o formato da mão foi representado por unidades do sistema numérico e alfabeto manual da ASL. Vários pesquisadores do mundo todo reforçaram o trabalho original de Stokoe e acharam uma nova disciplina de Linguística de Linguagem Gestual. Por isso, esse sistema não deve ser visto como uma simples notação, mas como uma “família” de sistemas relacionados (MARTIN, 2004).

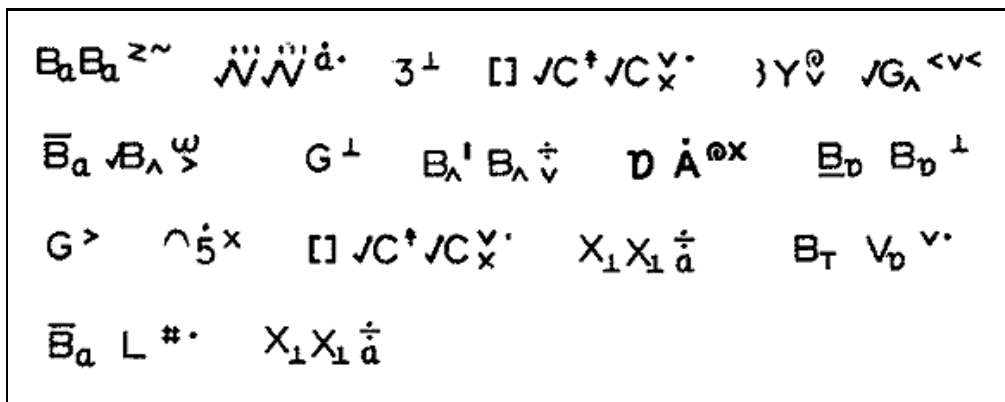


Figura 4.1: Exemplos da Notação de Stokoe

O HamNoSys foi desenvolvido, por um grupo de pessoas ouvintes e pessoas surdas, como uma ferramenta de pesquisa e sua primeira versão foi publicamente disponibilizada em 1989. O propósito do HamNoSys, ao contrário do *SignWriting*, nunca foi o uso da comunicação cotidiana em linguagem gestual. Foi projetado para ser usado em pesquisa de linguística computacional e deve ser aplicável a qualquer linguagem gestual no mundo (BENTELE, 2004). Consiste em aproximadamente 200 símbolos cobrindo os parâmetros de formato de mão, configuração de mão, posição e movimento conforme Figura 4.2.

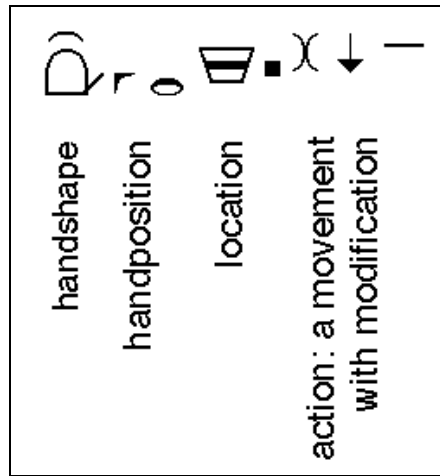


Figura 4.2: Exemplos da Notação HamNoSys (GERMAN SIGN LANGUAGE; DEAF, 2004)

## 4.1 SignWriting

Foi criado há cerca de 20 anos por Valerie Sutton, membro do DAC (Deaf Action Committee), um sistema de escrita para linguagens de sinais, chamado *SignWriting* (SW). As expressões gráficas de *SignWriting* (Figura 4.3) se restringem a descrever movimentos (manuais, braçais, assim como expressões faciais), fazendo com que o sistema possa, por isso, representar qualquer linguagem de sinais. Foi desenvolvido para ser uma forma escrita para línguas de sinais, assim como as diversas notações alfabéticas, silábicas e ideográficas foram desenvolvidas para serem formas escritas de línguas faladas. Uma diferença óbvia e crucial, porém, é que essas últimas tiveram um desenvolvimento de caráter histórico, ao passo que *SignWriting* foi racionalmente concebido, sendo, portanto uma linguagem formal (embora sua especificação tenha um caráter semi-formalizado, como acontece usualmente com a maioria das notações gráficas). Além do caráter gráfico, *SignWriting* foi concebido para ser registrado de modo estático, em papel. Isso se reflete no seu amplo repertório de símbolos dedicados à representação da dinâmica dos elementos corporais em jogo dentro dos sinais (COSTA, 2006e).



Figura 4.3: Alguns símbolos do *SignWriting*











Abaixo serão mostradas as características dos símbolos do SW, como é estruturado um dicionário de língua de sinais a partir do SignSpelling(SS) e do Sinal-Símbolo-Sequência(SSS), e os bancos de dados de sinais. Também serão mostrados como exemplos de editores de sinais que fazem uso dos símbolos do SW e a linguagem SWML (*SignWriting Markup Language*), que suporta o armazenamento, troca e processamento de textos do SW. E por fim o uso do SW no Brasil.

### 4.1.1 Símbolos SW

Atualmente os símbolos do *SignWriting* estão divididos em 10 categorias e 55 grupos conforme a configuração das mãos, contato das mãos, faces, movimentos do corpo e da

cabeça, ombro, membros, inclinação da cabeça, localização, movimento de dinâmicas e pontuação, conforme Tabela 4.1.1.

Tabela 4.1: Categorias SW

|   |                          |
|---|--------------------------|
|    | Category 1: Hand         |
|    | Category 2: Movement     |
|    | Category 3: Face         |
|    | Category 4: Head         |
|    | Category 5: Upper-Body   |
|    | Category 6: Limb         |
|  | Category 7: Full-Body    |
|  | Category 8: Location     |
|  | Category 9: Dynamics     |
|  | Category 10: Punctuation |

#### 4.1.2 Sinal-Símbolo-Sequência e SignSpelling

Os dicionários para línguas de sinais, tal como a Língua Americana de Sinais (ASL), não usa o alfabeto romano para classificar sinais. Os dicionários da língua de sinais são classificados por símbolos do *SignWriting*, de duas maneiras diferentes. Primeiramente, cada sinal tem uma soletração estabelecida de ordem dos símbolos do *SignWriting* dentro de um sinal. Isto é chamado um SignSpelling (SS). Mas os SS não são lineares, isto é, não são escritos da esquerda para a direita. Os símbolos estão em conjuntos visuais. O leitor lê o conjunto, ou a unidade, como um inteiro, mas há uma ordem sequencial a ser lida, e essa ordem sequencial transforma-se em SS. Uma vez que cada sinal tem seu próprio SignSpelling original, a seguir todos os sinais são classificados pelo Sinal-

Símbolo-Sequência (SSS), que coloca os símbolos em grupos. Os grupos são distinguidos pelo SSS. Os sinais podem ser encontrados no dicionário através do formato da mão que começa o sinal, na sequência do SSS (DAC, 2004a).

Ambos, SS e SSS, são necessários para criar os dicionários classificados por sinais. A seguir a diferença entre eles:

1. SignSpellings (SS) - ordem de símbolos do *SignWriting* dentro de um sinal. SignSpellings divide cada sinal em Sinal-Sílabas, que estabelecem em cada sinal a ordem dos conjuntos de símbolos.
2. Sinal-Símbolo-Sequência (SSS)- a lista inteira dos símbolos de *SignWriting* colocados em uma ordem específica, de modo que uma vez que um SignSpelling é estabelecido, o dicionário possa classificar aqueles sinais em grupos (Tabela 2), de modo que os usuários do dicionário possam encontrar os sinais através do formato das mãos e de outros detalhes.

Tabela 4.2: SSS de um símbolo SW

| Símbolo | Categoria | Grupo | N_Símbolo | Variação | Preenchimento | Rotação |
|---------|-----------|-------|-----------|----------|---------------|---------|
| *       | 02        | 01    | 001       | 01       | 01            | 01      |

*SignWriting* tem conjuntos de símbolos que dão forma a uma unidade visual. Estas unidades visuais têm uma sequência, mas não se escreve da esquerda para a direita. Portanto, às vezes o símbolo que é o primeiro no sinal, pode ser colocado debaixo da face. Às vezes as mãos começam no fundo do sinal, seguindo as setas ascendentes até o alto do conjunto, e outras vezes pode começar no centro do sinal, movendo-se para fora, etc. Assim, mesmo que os movimentos de *SignWriting* sejam sequenciais, não são amarrados visualmente da esquerda para a direita, como em línguas faladas (DAC, 2004a).

### 4.1.3 SymbolBank e SignSpelling

No SignBank (SUTTON, 2006), que um banco de dados de sinais, existem duas bases de dados diferentes que trabalham juntas para criar um dicionário Sinal-Palavra. O SSS é estabelecido pela base de dados de SymbolBank, mas SignSpellings são estabelecidos na base de dados de SignSpellings. As duas bases de dados podem ser vistas no Editor SignBank, em duas colunas separadas, como mostra a Figura 4.4.

Um SignSpelling consiste de uma série de sílabas de *SignWriting* que são escritas sequencialmente no tempo. A sílaba 1 vem sempre antes da sílaba 2. Mas dentro de cada sílaba, o tempo continua o mesmo. Os símbolos dentro de uma sílaba "estão acontecendo ao mesmo tempo". Então, SignSpellings é simultâneo e sequencial, como mostrado na Figura 4.5

Cada sílaba de *SignWriting* tem sua própria ordem da soletração. As sílabas 1 e 3 são sílabas da mão. A sílaba 2 é uma sílaba de movimento.

Pode-se escolher soletrar com simplicidade ou detalhadamente, dependendo do tamanho de seu dicionário. Um dicionário grande requer uma soletração mais detalhada.

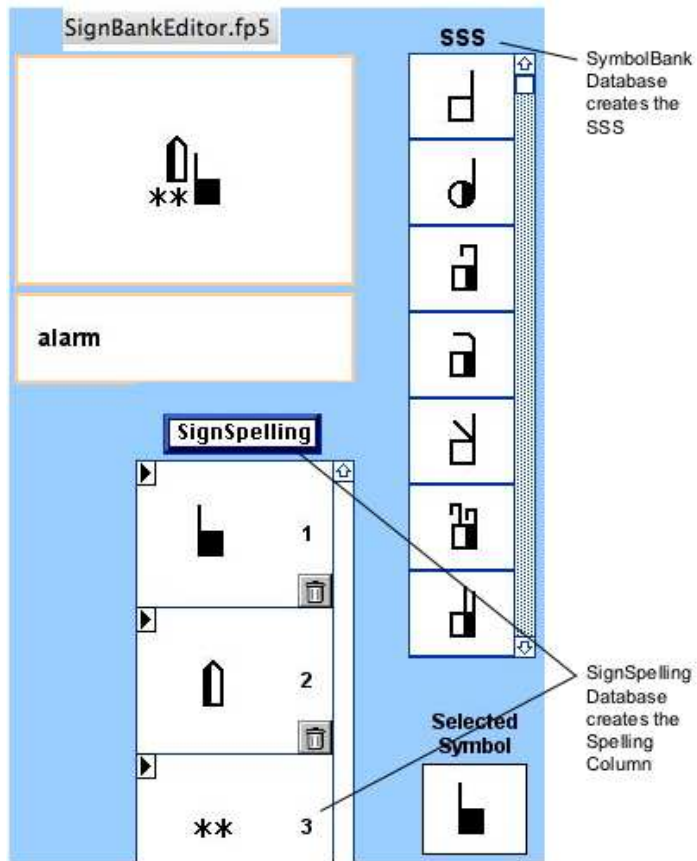


Figura 4.4: Editor de Sinais SignBank (DAC, 2004a)

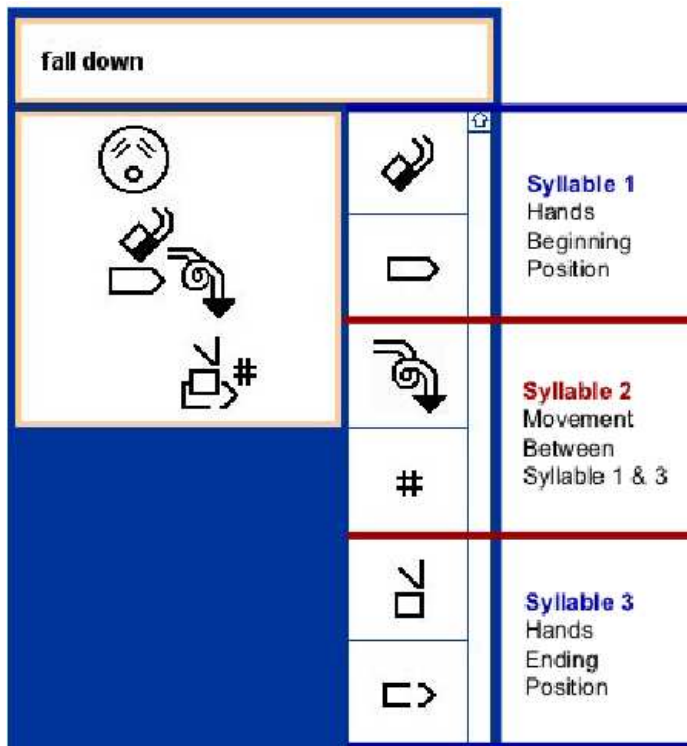


Figura 4.5: Divisão de Sílabas em *SignWriting* (DAC, 2004a)

Soletrações simples (Figura 4.6):

Sílabas das Mãos:

1. Mão Dominante
2. Mão Não-Dominante

Sílabas de Movimento:

1. Movimento da mão dominante
2. Movimento da mão não-dominante

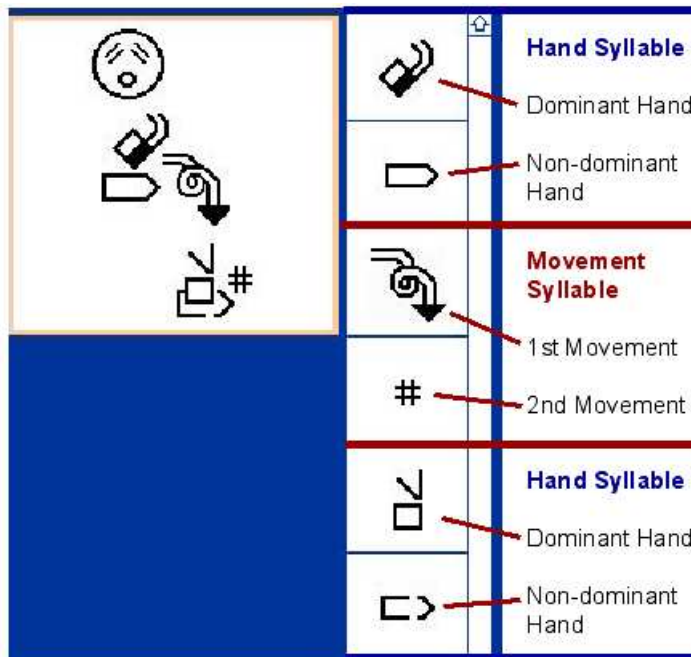


Figura 4.6: Soletração básica (DAC, 2004a)

Soletrações detalhadas (Figura 4.7):

Sílabas das Mãos:

1. Mão Dominante
2. Mão Não-Dominante
3. Localização Dominante
4. Localização Não-Dominante
5. Posição dos braços
6. Posição dos Ombros
7. Posição da Cabeça
8. Expressão Facial
9. Localização da Sílaba

Sílabas de Movimento:

1. 1<sup>o</sup> movimento da mão dominante
2. 1<sup>o</sup> movimento da mão não-dominante
3. 2<sup>o</sup> movimento dominante
4. 2<sup>o</sup> movimento não-dominante
5. Localização da Sílaba

A maioria das soletrações não são nem detalhadas nem básicas, mas procura-se um meio termo entre as duas, sem detalhes em demasia, apenas o suficiente para diferenciar sinais similares.



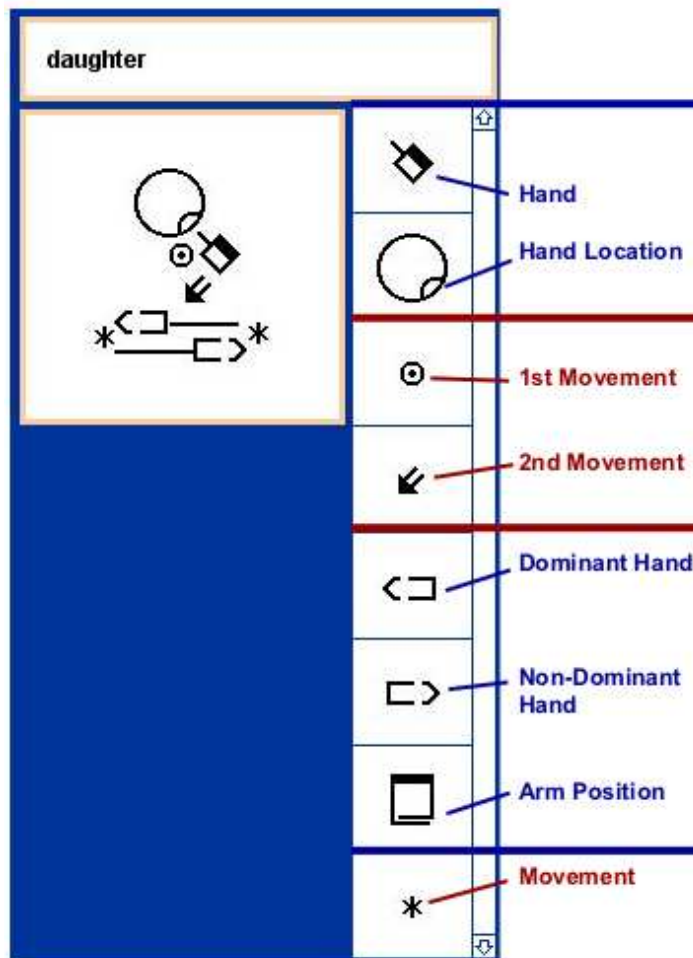


Figura 4.7: Soletação detalhada (DAC, 2004a)

No exemplo abaixo, o marcador de posição é a orelha, e é uma parte necessária da soletação, já que dá significado ao sinal. Quando a localização dá o significado ao sinal ele deve ser incluído. Alguns casos de soletação podem ser vistos no Anexo A.



Figura 4.8: Sinal “surdo”

#### 4.1.4 Editores

Existem alguns editores de sinais que fazem uso dos símbolos *SW*, alguns deles disponíveis eletronicamente pelo DAC (DAC, 2004b).

##### 1. SignWriter

SignWriter é o primeiro editor de linguagens de sinais do mundo, processando sinais de maneira visual com símbolos do *SW*. O SignWriter possibilita aos usuários da linguagem de sinais poderem escrever textos na sua própria linguagem. Dispõe

de símbolos de movimentos, formato de mãos, expressões faciais e ainda um Dicionário de Linguagens de Sinais, onde os usuários podem armazenar sinais, copiá-los e colá-los em documentos enquanto digitam.

Com o SignWriter, o usuário tem acesso a arquivos de dicionários de 17 países e mensagens na tela estão disponíveis em 10 línguas faladas.

O SignWriter está disponível em duas versões: DOS (DAC, 2004b) e JAVA (DAC, 2004c). O SignWriter DOS, em MS-DOS (Figura 4.9), está disponível para Windows 95, 98, 98SE ou ME. O SignWriter Java, em Java2 v1.4.1 está disponível para Macintosh ou Windows. Atualmente, estão em andamento as versões em Python e Tiger.



Figura 4.9: Editor Sign Writer

## 2. SWEdit

Nesta seção apresenta-se o editor para textos em línguas de sinais escritos em *SignWriting*, o SWEdit, desenvolvido na UCPEl<sup>1</sup> (TORCHELSEN; COSTA; DIMURO, 2002), e motivado pela necessidade de se fornecer uma ferramenta para auxiliar pessoas surdas a terem uma melhor comunicação através do computador e, pela escassez de programas que utilizam a linguagem da comunidade surda. É o editor usado para a produção de sinais deste trabalho.

Os requisitos para se obter o editor eram:

- Edição de textos usando o sistema *SignWriting*
- Inclusão de textos em linguagem oral
- Inclusão de figuras e imagens
- Drag & Drop entre diferentes programas
- Base de dados expansível
- Dicionários
- Salvar e carregar arquivos no formato SWML

<sup>1</sup>Universidade Católica de Pelotas

- Interface projetada especialmente para a utilização por pessoas surdas

O SWEdit é executado nos sistemas Windows95/98/ME/XP/NT/2000, UNIX e MAC OS, devida a utilização da biblioteca gráfica wxWindows e por não fazer uso de chamadas de sistema(System Calls) e não fazer uso de recursos dependentes de sistema.

SWEdit (Figura 4.10) foi desenvolvido pensando-se nas dificuldades e qualidades de portadores de deficiência auditiva. A principal preocupação foi a dificuldade de leitura de textos na forma oral, ocasionando a substituição e utilização de textos por figuras e imagens para a representação de ações ou ferramentas do editor. A qualidade mais explorada no editor é a capacidade de interpretação visual da comunidade. Essa qualidade foi explorada utilizando-se figuras auto-explicativas, as quais aparecem em botões e menus.

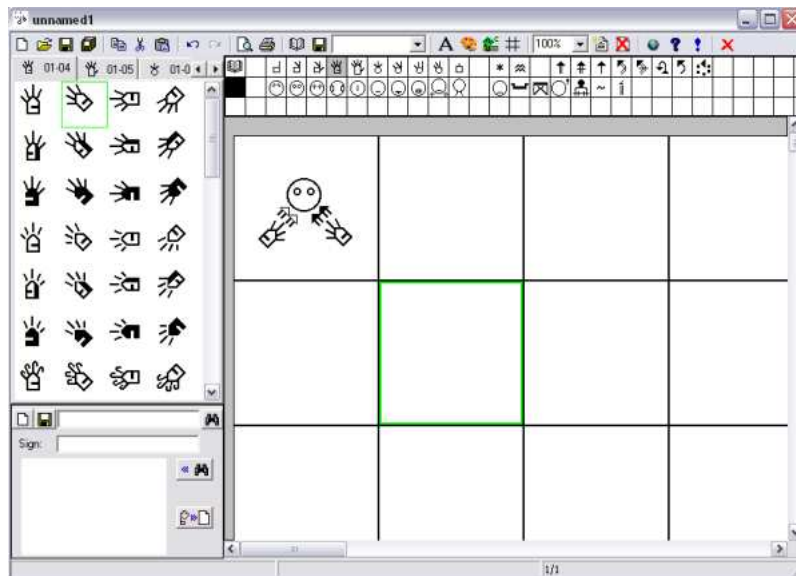


Figura 4.10: Editor SWEdit

## 4.2 SWML

SWML (COSTA, 2006a) é uma linguagem de marcação baseada em XML, que suporta adequadamente o armazenamento, troca e processamento de textos do *SignWriting*. Com a SWML é possível o intercâmbio de documentos entre diferentes programas, a análise de textos independente do editor e também serve como um formato de armazenamento de textos (Figura 4.11).

A SWML torna possível a conversão<sup>2</sup> de arquivos de sinais em arquivos texto. Um sinal convertido em textos SWML apresenta a posição, rotação, variação, preenchimento e forma de cada símbolo, conforme Figura 4.12. Um exemplo de documento SWML completo pode ser visto no Anexo B.

<sup>2</sup>Conversor SWML está disponível em <http://swml.ucpel.tche.br>

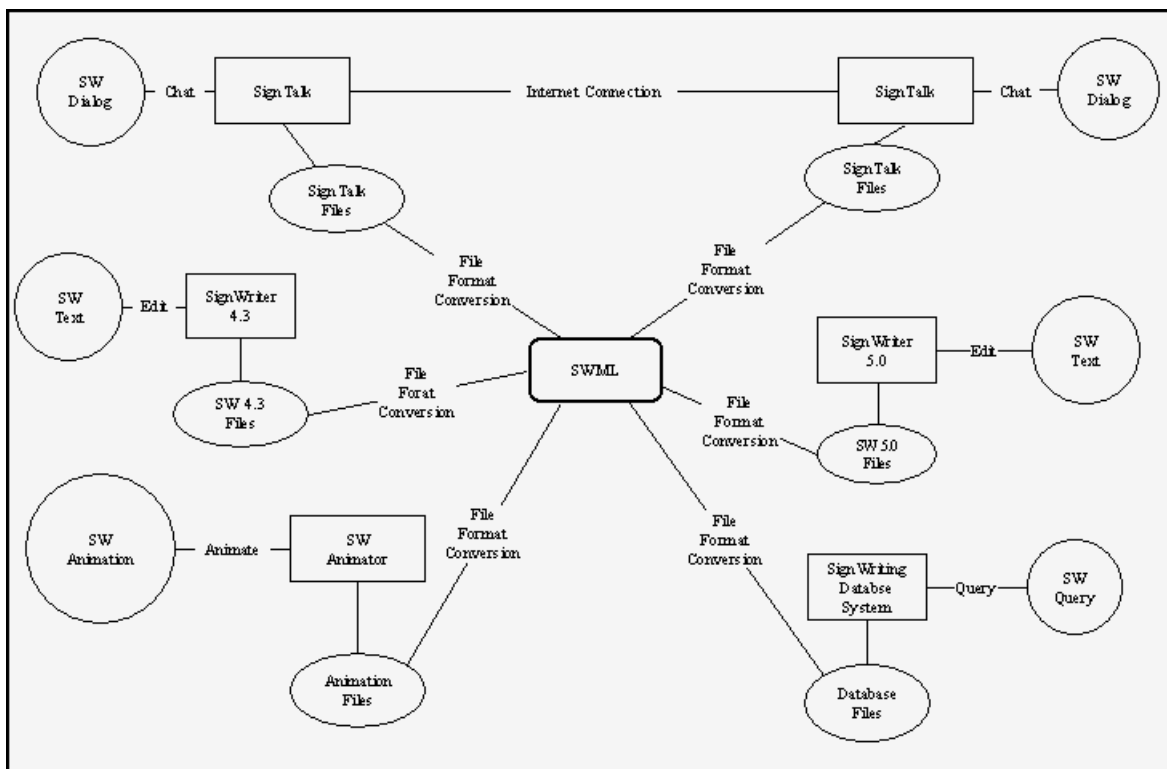


Figura 4.11: Intercâmbio de arquivos SW entre diferentes aplicações (COSTA, 2006a)

```

<symb x="61" y="90" x-flop="1" y-flop="0" color="0,0,0">
  <category>02</category>
  <group>03</group>
  <symbnum>001</symbnum>
  <variation>01</variation>
  <fill>01</fill>
  <rotation>01</rotation>
</symb>

```

Figura 4.12: Trecho de um texto SWML

### 4.3 Uso do *SignWriting* no Brasil

O *SignWriting* é usado em mais de 30 países e no Brasil começou a ser utilizado em 1996. A partir disso se formaram grupos de trabalho com pessoas ouvintes e deficientes auditivas para a difusão do sistema e alfabetização da comunidade surda. Formaram-se também grupos de pesquisa para auxiliar no desenvolvimento de ferramentas que façam uso do sistema, entre as quais se encaixa o sistema descrito no presente trabalho.

Stumpf (STUMPF, 2000), relatando sua experiência sobre o uso do *SignWriting* na Escola Especial Concórdia, afirma que muitos alunos surdos quando aprendem a escrever pensam que o português escrito é a representação da língua de sinais que eles usam. Quando os alunos começam a aprender a escrita de sinais eles conseguem separar e ver que é outra língua. Trabalhando as duas línguas separadamente e comparando-as o resultado será melhor porque é assim que acontece a aprendizagem de uma segunda língua.

Os surdos em suas manifestações escritas podem produzir ótimos materiais como: literatura, poesia e textos, se possuírem o domínio do instrumento necessário para isso. Para

a maioria, isto não acontece na língua portuguesa porque a dificuldade de aprendizado da língua oral pelos surdos é imensa.

Por esse motivo sempre existiu um grande interesse por parte dos surdos em aprender a língua de sinais e, mais recentemente, em usar o sistema *SignWriting*.

Relata-se aqui alguns exemplos da utilização do *SignWriting* no desenvolvimento de ferramentas que venham a facilitar o processo de ensino e aprendizagem da comunidade surda.

Na USP, sob a coordenação do Prof. Dr. Fernando Capovilla, e participação de informantes surdos da COPAVI (Cooperativa Padre Vicente - SP) e da FENEIS, foram preparados dois dicionários de língua brasileira de sinais (CAPOVILLA et al., 1999).

Na UCPel, o grupo de pesquisa que coordena o projeto SignNet (COSTA, 2006c) desenvolve textos em *SignWriting* e desenvolve a infraestrutura de *software* mínima necessária para viabilizar a aplicação das técnicas de processamento de linguagens naturais (que vem sendo desenvolvidas, há muito, nas áreas da Inteligência Artificial e da Linguística Computacional) às línguas de sinais utilizadas pelos surdos. Em especial, visa fazer isso para o caso da LIBRAS.

Na PUC-RS foi desenvolvido o SIGNSIM (CAMPOS; GIRAFFA; SANTAROSA, 2000), uma ferramenta para o auxílio da aprendizagem da Língua de Sinais tanto para surdos quanto para ouvintes e, apresenta-se como um sistema de tradução entre a escrita da Língua Portuguesa e a escrita da Língua de Sinais.

## 5 AGA

Com o intuito de produzir-se as animações de sinais, estuda-se o Modelo AGA (Animação Gráfica baseada em Autômatos Finitos) (ACCORSI, 2002). Neste capítulo são apresentados seus conceitos e suas características.

O modelo AGA especifica a animação a partir de um conjunto de atores (objetos) e suas respectivas variações durante a animação. As especificações em AGA, entretanto, são suportadas por um modelo formal baseado em autômatos com saída (HOPCROFT; ULLMAN, 2000; MENEZES, 2005).

No AGA, cada um dos atores na animação é especificado a partir de uma extensão proposta para o autômato com saída, a qual vincula as variações estéticas do ator à saída do autômato. Deste modo, quando os autômatos são simulados, mediante a leitura da fita de entrada, as transições entre seus estados controlam a animação dos atores (ACCORSI, 2002).

A Figura 5.1 ilustra a estrutura básica de especificação do modelo AGA aplicada a uma animação com três atores. Os atores cabeça, mão direita e mão esquerda são especificados a partir de autômatos com saída onde representações gráficas estão associadas às transições. Essas representações correspondem às variações gráficas que o ator pode sofrer durante a animação. Desta forma, quando o ator mão direita realiza uma transição de estado, a representação gráfica do ator apresentada na animação é alterada conforme as informações da fita.

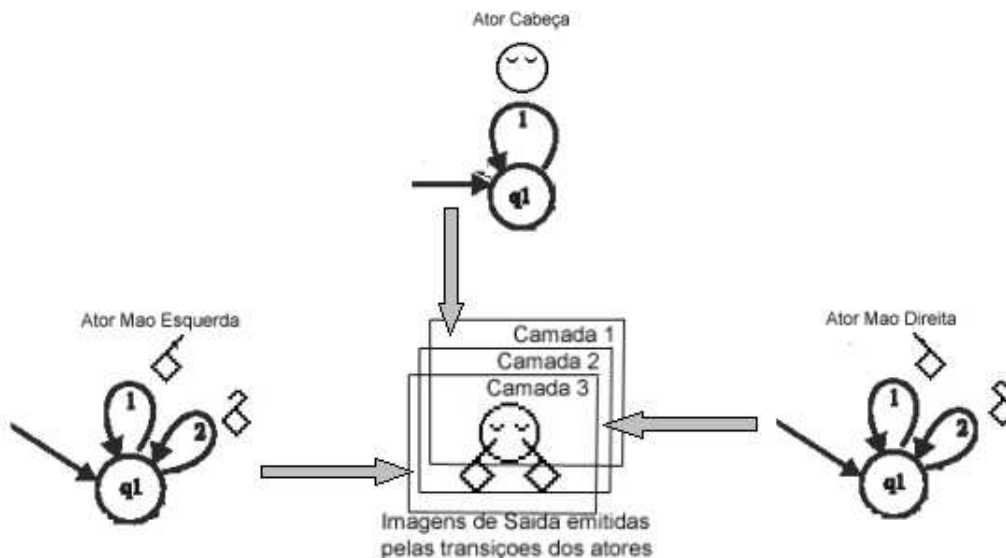


Figura 5.1: Atores especificados no modelo AGA

A fita de entrada utilizada pelo autômato contém, além dos símbolos de entrada, especificações complementares sobre os instantes em que devem ocorrer as transições, assim como funções de controle para a saída do autômato (Figura 5.2). Essas funções de controle são responsáveis por diferentes tarefas, como por exemplo, posicionar o ator na área de animação. A área de atuação dos atores pode ser dividida em camadas com o objetivo de estabelecer a ordem de sobreposição das figuras de saída. Assim, as cenas correspondentes a cada instante da animação são formadas pela unificação das camadas (ACCORSI, 2002).

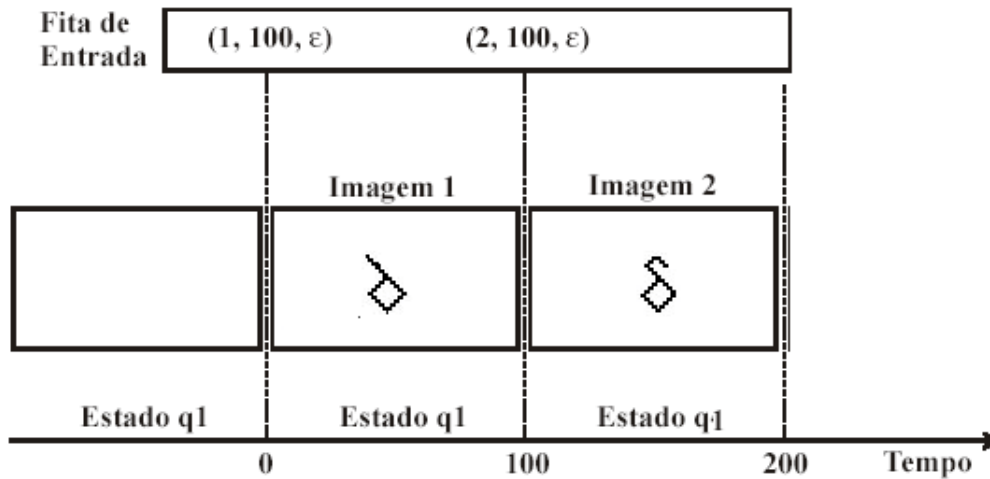


Figura 5.2: Troca de imagens do ator mão direita provocadas pela leitura da fita

## 5.1 Modelo AGA

A estrutura do modelo AGA é baseada em uma unidade básica, chamada de ator AGA, a qual é definida a partir de extensões propostas para os autômatos com saída. Essas extensões complementam a definição de autômato com saída como o objetivo de suportar o controle temporal das transições, prover imagens como saída com adaptações contextuais e associar descrições semânticas aos estados (ACCORSI, 2002).

### 5.1.1 Autômatos Finitos com Saída

O conceito básico de autômato finito limita a saída a uma informação binária: aceita ou rejeita. Porém, há duas abordagens possíveis: Máquina de Mealy e Máquina de Moore (HOPCROFT; ULLMAN, 2000; MENEZES, 2005), que estendem esse conceito básico para modelos que possibilitam a geração de palavras como saída. A Máquina de Mealy associa as saídas às transições, enquanto a Máquina de Moore aos estados.

Para a apresentação e implementação do modelo, preferiu-se partir da Máquina de Mealy, pois associa as imagens de saída às transições, os estados podem ser incluídos de acordo com a modelagem desejada para o ator, independente do número de imagens no alfabeto de saída. As descrições semânticas associadas aos estados também não são vinculadas à mesma imagem, pois várias transições com imagens de saída diferentes podem ser mapeadas para o mesmo estado. Essas características proporcionam uma flexibilidade maior para a modelagem dos atores (ACCORSI, 2002).

Assim, para as extensões propostas à frente, a Máquina de Mealy (HOPCROFT; ULLMAN, 2000; MENEZES, 2005), utilizada como base, é um Autômato Finito De-

terminístico com saídas associadas às transições e representada pela 6-upla,  $ME = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ , onde:

- $Q$  conjunto finito de estados possíveis do autômato;
- $\Sigma$  alfabeto de símbolos de entrada;
- $\Delta$  alfabeto de símbolos de saída;
- $\delta$  função programa ou função de transição (função parcial  $\delta : Q \times \Sigma \leftarrow Q$ );
- $\lambda$  função saída (função parcial  $\lambda : Q \times \Sigma \leftarrow \Delta^*$ );
- $q_0$  estado inicial do autômato ( $q_0 \in Q$ ).

O processamento de uma Máquina de Mealy, para uma palavra de entrada  $w$ , consiste na sucessiva aplicação da função programa para cada símbolo de  $w$  até ocorrer uma condição de parada. A medida que cada símbolo de  $w$  é lido, a palavra, vinculada como saída à transição efetuada, é inserida na fita de saída. A palavra vazia como saída da função  $\lambda$  indica que nenhuma gravação é realizada na fita de saída.

### 5.1.2 Função de Saída Contextual

A primeira extensão para a Máquina de Mealy é realizada quanto a saída do autômato, com o objetivo de adequar a saída à emissão de imagens, de tal forma que estas possam ser contextualizadas a diversas situações durante a animação.

Com isso, o alfabeto de saída  $\Delta$ , para o ator AGA, é definido como um conjunto finito de imagens estáticas, onde cada elemento é uma imagem distinta que o ator pode utilizar durante a apresentação. O conjunto  $\Delta^*$ , portanto, é composto por todas as imagens que podem ser produzidas pela composição de zero ou mais imagens de  $\Delta$ . A operação de composição pode ser definida de maneira mais conveniente à classe de animações representadas. Para a elaboração dos exemplos e a implementação do modelo, a palavra  $i_1, i_2, \dots, i_n \in \Delta^*$  representa a composição  $i_1 + i_2 + \dots + i_n = i_r$  que produz a imagem  $i_r$  a partir da sobreposição da imagem  $i_1$  pela  $i_2$  e assim sucessivamente até  $i_n$ .

Dependendo do instante da animação, a imagem de saída pode sofrer transformações com o objetivo de se adequar ao contexto da cena representada. Por exemplo, a mesma imagem pode ser apresentada em posições ou escalas diferentes. Para esse propósito, é introduzido o conjunto  $F$ , cujos elementos são funções do tipo  $f(\vec{v}, i) = i^t$ , onde  $i$  e  $i^t$  são elementos de  $\Delta^F$  e  $\vec{v} = v_1 v_2 \dots v_n$  é um vetor de argumentos necessários para o mapeamento de  $i$  para  $i^t$ . O conjunto  $\Delta^F$  é composto por todas as imagens de  $\Delta^*$  transformadas por zero ou mais funções.

Assim, a função de saída estendida, chamada de função de saída contextual, é definida como a função parcial  $\lambda_c : Q \times \Sigma \times F^* \leftarrow \Delta^F$ , onde  $F^*$  é o conjunto de cadeias  $\varphi = f_1 f_2 \dots f_m$  formadas por zero ou mais funções de  $F$ . Logo  $\lambda(q, a, \varphi) = f_1(\vec{v}_1, f_2(\vec{v}_2, \dots, f_m(\vec{v}_m, i)))$ .

### 5.1.3 Função de Transição Temporal

Como as variações que os atores podem sofrer durante a animação estão associadas às transições do autômato, é preciso determinar em que momentos essas variações deverão ocorrer durante o processo de avaliação da fita de entrada. Então, cada símbolo da entrada é associado a um número de natural que determina o tempo de espera, em milésimos de segundos, após a leitura de um símbolo. Desta maneira, a fita de entrada  $f_t = (a_1, t_1, \varphi_1)(a_2, t_2, \varphi_2) \dots (a_n, t_n, \varphi_n)$  possui os triplos ordenados  $(a, t, \varphi) \in \Sigma \times N \times F^*$ , onde  $a$  é um símbolo do alfabeto de entrada,  $t$  é um tempo em milésimos de segundo e  $\varphi$  uma cadeia de funções de transformação da saída.



Para o comportamento temporal do autômato, as transições são consideradas instantâneas e o tempo passa durante a ocupação de um estado. Assim, para a entrada  $(a_1, t_1, \varphi_1)(a_2, t_2, \varphi_2) \dots (a_n, t_n, \varphi_n)$  que provoca a sequência de ocupação dos estados  $q_0 q_1 \dots q_n$ , o tempo  $t_k$ , para algum  $1 \leq k \leq n$ , determina o tempo de ocupação do estado  $q_k$ .

Deste modo, a função de transição é estendida para comportar um argumento adicional que determina o tempo que deve ser esperado após ser efetuada a transição. Esse argumento é obtido a cada célula lida a partir da fita de entrada. Portanto, a função de transição temporal é definida como a função parcial  $\delta_t : Q \times \Sigma \times N \leftarrow Q$ , para a qual, a especificação  $\delta_t(q, a, t) = q'$  determina que deve ocorrer uma transição de  $q$  para  $q'$  quando o símbolo  $a$  é lido, e depois, aguardando o tempo  $t$  no estado  $q'$  antes da leitura do próximo símbolo.

### 5.1.4 Função Descrição

A última extensão proposta pela máquina de Mealy é destinada à vinculação de descrições semânticas aos estados. A estrutura da especificação do modelo AGA baseada em conjunto de atores colabora para que o comportamento de cada ator possa ser verificado a partir de seu autômato correspondente. Desta forma, os estados dos autômatos podem ser utilizados para indicar diferentes situações durante a animação.

Com esse propósito, foi criada a função descrição que mapeia os estados de cada ator AGA para um conjunto de descrições. A forma de descrição semântica do estado não é definida neste texto, pois pode ser explorada de diferentes maneiras dependendo do mecanismo de consulta. Contudo, para os exemplos foram empregadas descrições baseadas em palavras-chaves.

A função descrição é definida como a função parcial  $\sigma : Q \leftarrow D$ , onde  $D$  é o conjunto de descrições utilizadas pelo ator AGA especificado. Nem todo estado precisa necessariamente estar mapeado para alguma descrição em  $D$ . Por exemplo, para o ator a esquerda na Figura 5.3, os seguintes mapeamentos são definidos para o conjunto  $D = \{ \text{"Dedo esticado"}, \text{"Dedo dobrado"} \}$ :  $\sigma(q1) = \text{"Dedo esticado"}$  e  $\sigma(q2) = \text{"Dedo dobrado"}$ . Desta maneira, toda vez que o ator A ocupar o estado 2 durante a animação e sabido que o ator está na situação de "Dedo dobrado".



Figura 5.3: Alternativas de construção dos atores AGA

## 5.2 Implementação do Modelo AGA

A abordagem utilizada na implementação do AGA é baseada na produção de animações que executem em tempo real. A estratégia adotada é semelhante às empregadas pelo

Flash e o SVG, nas quais os quadros são gerados no momento da apresentação a partir do processamento de um arquivo de intercâmbio contendo as especificações da animação.

Nas animações desenvolvidas em AGA, o arquivo de intercâmbio contém as especificações dos atores AGA e suas respectivas fitas de entrada descritas em AgaML. Essa linguagem é um vocabulário XML desenvolvido especificamente para esse propósito. Os arquivos em AgaML são disponibilizados pelo servidor HTTP juntamente com as imagens e sons utilizados como saída pelos atores AGA.

A carga do arquivo de intercâmbio e a geração dos quadros são realizadas pelo programa de visualização, AGA Player, executado no navegador do cliente. O AGA Player, desenvolvido em linguagem Java na forma de applet, é carregado pelo cliente quando alguma página Web que o vincula é processada pelo navegador. Como o applet desenvolvido utiliza recursos da plataforma Java que os navegadores ainda não suportam totalmente (processamento de documentos em XML), é necessário que o cliente tenha instalado adicionalmente o Java plug-in, fornecido pela Sun Microsystems, para processá-lo. Este plug-in basicamente redireciona a interpretação do applet para JRE (Java Runtime Environment) instalada no cliente que provê esses recursos adicionais.

A animação a ser reproduzida é associada à página Web através de um dos parâmetros fornecidos ao applet. Assim, quando o AGA Player é executado, o arquivo em AgaML atribuído ao parâmetro é carregado juntamente com as mídias externas e processado produzindo a animação.

O applet possui outro parâmetro que determina se a animação deve ser reproduzida com ou sem o painel de controle. Este recurso favorece tanto a criação de animações simples de aplicação direta nas páginas, como o caso dos banners de propaganda, quanto animações mais complexas, as quais o observador precisa explorar o conteúdo da seqüência animada.

### 5.2.1 Linguagem AgaML

A AgaML (AGA Markup Language) é um vocabulário XML desenvolvido especialmente para especificar as animações no modelo AGA. Duas características do XML motivaram sua escolha para o desenvolvimento da AgaML: sua ampla aplicabilidade como padrão de intercâmbio de documentos na Web; e a facilidade de manipulação tanto manual quanto por processos automatizados. A segunda característica colabora para que a AgaML seja utilizada tanto como linguagem fonte para edição direta pelo usuário quanto em ambientes automatizados de animação na qualidade de linguagem intermediária.

O desenvolvimento da linguagem AgaML teve como foco os seguintes requisitos:

- prover uma estrutura sintática tanto para a especificação dos atores como para as fitas de entrada seguindo a definição formal proposta pelo modelo AGA;
- possibilitar a utilização da mesma especificação de ator ou fita para a criação de múltiplas instâncias na animação;
- agrupar as especificações dos atores em unidades sintáticas coesas de tal modo a possibilitar sua fácil reutilização e compartilhamento pelas facilidades herdadas do XML;
- integrar símbolos do alfabeto de saída definidos externamente, com o objetivo de utilizar imagens e sons em diferentes formatos e localizações;

- fornecer informações adicionais para catalogação da animação com o objetivo de favorecer mecanismos automáticos de consulta.

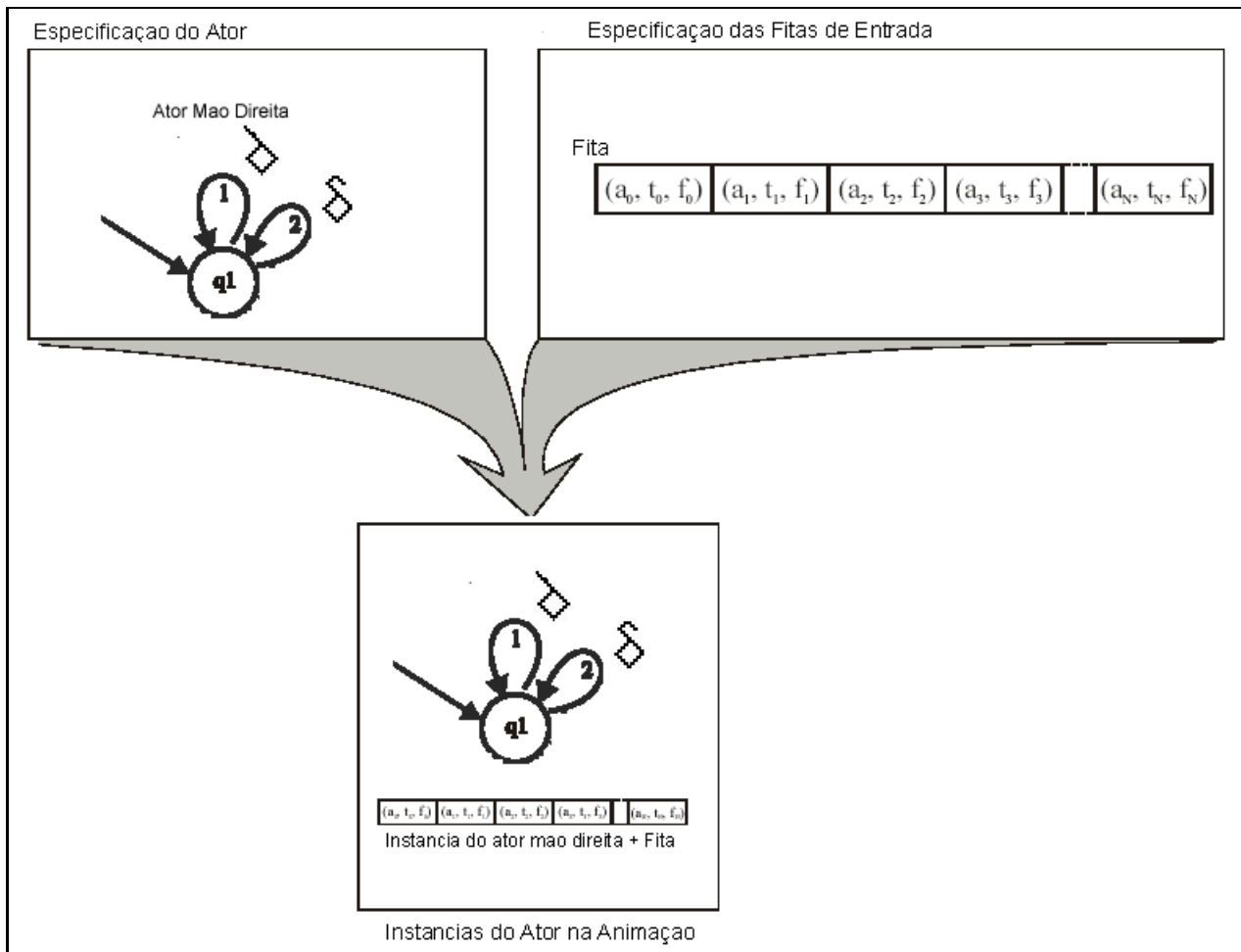


Figura 5.4: Estrutura de instânciação dos atores

AgaML organiza a especificação da animação a partir de três componentes básicos: a especificação dos atores AGA, a especificação das fitas de entrada e a criação das instâncias dos atores. Uma instância pode ser entendida como a associação da especificação de um ator AGA com uma fita de entrada, como ilustrado na Figura 5.4.

As seções seguintes descrevem os elementos definidos na DTD da AgaML acompanhados de exemplos aplicados. A DTD é utilizada como padrão para definição 39 das regras sintáticas de novos vocabulários XML.

### 5.2.2 Especificação da Animação

O elemento raiz AGA da DTD (Figura 5.5) comporta as especificações dos atores, das fitas e as definições das instâncias dos atores. Além disso, disponibiliza informações adicionais para auxiliar no processo de apresentação da animação.

Os atributos do elemento AGA: *WIDTH*, *HEIGHT*, *BACKGROUND*, *FRAMERATE* e *REPEAT*, descrevem, respectivamente, a largura e altura da janela onde será exibida a animação, a cor de fundo, o número de quadros por segundo e o número de vezes que a animação deve ser repetida. A palavra *LOOP* pode ser associada ao atributo *REPEAT* para provocar a repetição contínua da animação. O valor atribuído ao *FRAMERATE* determina

```

<IELEMENT AGA (HEAD, ACTOR+, TAPE*, INSTANCE*, BITAPE*, COMBINE*)>
<!ATTLIST AGA
  VERSION CDATA #FIXED "2.0"
  WIDTH CDATA #REQUIRED
  HEIGHT CDATA #REQUIRED
  BACKGROUND CDATA #REQUIRED
  FRAMERATE CDATA #REQUIRED
  REPEAT CDATA #IMPLIED>
<IELEMENT HEAD (TITLE, AUTHOR, SUBJECT)>
<IELEMENT TITLE (#PCDATA)>
<IELEMENT AUTHOR (#PCDATA)>
<IELEMENT SUBJECT (#PCDATA)>

```

Figura 5.5: Trecho da DTD com os elementos *AGA*, *HEAD*, *TITLE*, *AUTHOR* e *SUBJECT*

a frequência com que os autômatos serão verificados pelo AGA Player para a formação dos quadros de saída. É recomendado que o valor do *FRAMERATE* seja compatível com as temporizações especificadas na fita de entrada, pois se o intervalo entre a composição de um quadro e outro for superior ao intervalo entre as transições, nem todas mudanças das imagens de saída serão observadas.

### 5.2.3 Especificação do Ator AGA

O *ACTOR*, elemento criado para este propósito, contém elementos filhos que descrevem o alfabeto de saída  $\Delta$ (*OUTPUT*), a função descrição  $\sigma$ (*DESCF*) e a função transição temporal  $\delta_t$  (*TRANSF*) (figura 5.6). Os atributos *ID*, *TYPE*, *STATES* e *SYMBOLS* dessa estrutura comportam, respectivamente, as informações sobre a identificação do ator, o seu tipo, as quantidades de estados e símbolos do seu alfabeto de entrada. A identificação é utilizada na criação da instância para a associação da especificação do ator com a fita de entrada. O tipo determina se o ator possui imagens (*GRAPHIC*) ou sons (*SOUND*) como alfabeto de saída associado.

A integração das imagens e sons do alfabeto de saída é realizada através do elemento *OUTPUT*, o qual define uma identificação interna para o símbolo (*ID*) e a associa com o arquivo que contém a imagem ou som (*SOURCE*). Os atributos *X* e *Y* representam os deslocamentos que devem ser aplicados à imagem para o enquadramento com as demais.

É importante notar que o elemento *OUTPUT* possibilita independência entre a linguagem AgaML e o formato de codificação de cada mídia. Novos formatos de mídia podem ser utilizados a partir da implementação dos respectivos decodificadores no AGA Player.

A função descrição, definida no modelo AGA, é representada por *DESCF*. Os elementos filhos *DESCRIPTION* associam o estado do ator AGA (atributo *STATE*) com um texto. Neste texto podem ser incluídas várias palavras que descrevam semanticamente a situação do ator naquele estado, como ilustrado na Figura 5.7.

As transições do ator e a vinculação com a saída são obtidas através do elemento *TRANSF*. Cada elemento filho *FROM* contém os arcos que divergem do estado (*STATE*) representados pelos elementos vazios *TO*. Os atributos de *TO* indicam o estado destino (*STATE*), o símbolo do alfabeto de entrada (*SYMBOL*) que provoca a transição, e a saída associada (*OUTPUT*). O atributo *OUTPUT* utiliza os identificadores definidos através do elemento *OUTPUT*. É possível definir a composição de imagens para saída a partir da enumeração das componentes em *OUTPUT*. Por exemplo, *OUTPUT= 1;2* provoca a sobreposição da imagem 1 com a 2 para produzir a imagem resultante de saída.

```

<IELEMENT ACTOR (OUTPUT*, DESCF?, TRANSF)>
<!ATTLIST ACTOR
    ID CDATA #REQUIRED
    TYPE CDATA #REQUIRED
    STATES CDATA #REQUIRED
    SYMBOLS CDATA #REQUIRED>
<IELEMENT OUTPUT EMPTY>
<!ATTLIST OUTPUT
    ID CDATA #REQUIRED
    SOURCE CDATA #REQUIRED
    X CDATA #REQUIRED
    Y CDATA #REQUIRED>
<IELEMENT DESCF (DESCRIPTION+)>
<IELEMENT DESCRIPTION (#PCDATA)>
<!ATTLIST DESCRIPTION
    STATE CDATA #REQUIRED>
<IELEMENT TRANSF (FROM+)>
<IELEMENT FROM (TO+)>
<!ATTLIST FROM
    STATE CDATA #REQUIRED>
<IELEMENT TO EMPTY>
<!ATTLIST TO
    STATE CDATA #REQUIRED
    SYMBOL CDATA #REQUIRED
    OUTPUT CDATA #IMPLIED>

```

Figura 5.6: Trecho da DTD com os elementos *ACTOR*, *OUTPUT*, *DESCF*, *DESCRIPTION*, *TRANSF*, *FROM* e *TO*

#### 5.2.4 Especificação da Fita de Entrada

A fita de entrada reúne as informações que descrevem o comportamento do ator durante a animação. O elemento *TAPE* (Figura 5.8) contém uma coleção de elementos vazios *CEL* que descrevem cada célula da fita. Como cada célula possui um símbolo de entrada (*SYMBOL*), um tempo de espera para a leitura do próximo símbolo (*TIME*) e a coleção de transformações que devem ser aplicadas à imagem de saída (*FN*). O valor de *TIME* é dado em milésimos de segundo. Alguns comportamentos em animações são repetitivos e conseqüentemente implicam na criação de células com o mesmo conteúdo. Devido a esta característica, o elemento *GROUP* foi definido com o propósito de agrupar células que irão se repetir, diminuindo assim, o tamanho da representação da fita. O número de repetições é determinado pelo atributo *ITERATION*. A definição da DTD permite que existam grupos dentro de grupos, o que favorece a representação de comportamentos repetitivos mais complexos. A Figura 5.9, por exemplo, ilustra uma fita de entrada criada para provocar o comportamento de andar no ator bicho. Nesta fita, duas células serão repetidas 3 vezes, totalizando assim, oito células a serem lidas pelo ator AGA.

Inicialmente foram definidas para a linguagem AgaML sete transformações que podem ser utilizadas no atributo *FN*. Transformações geométricas para rotação (*ROTATE*), translação (*TRANSLATE*), escala (*SCALE*), espelhamento (*FLIP*), além de oferecer a possibilidade de aplicar-se diretamente uma matriz homogênea de transformação (*MATRIX*). Como também transformações para o controle da visualização do ator, como a visibilidade (*VISIBLE*) e alteração de camada (*ORDER*).

Há dois símbolos especiais definidos para o atributo *SYMBOL*, o *WAIT* e o *EMPTY*. A presença do *WAIT* na célula indica que animação deve parar até que um evento externo

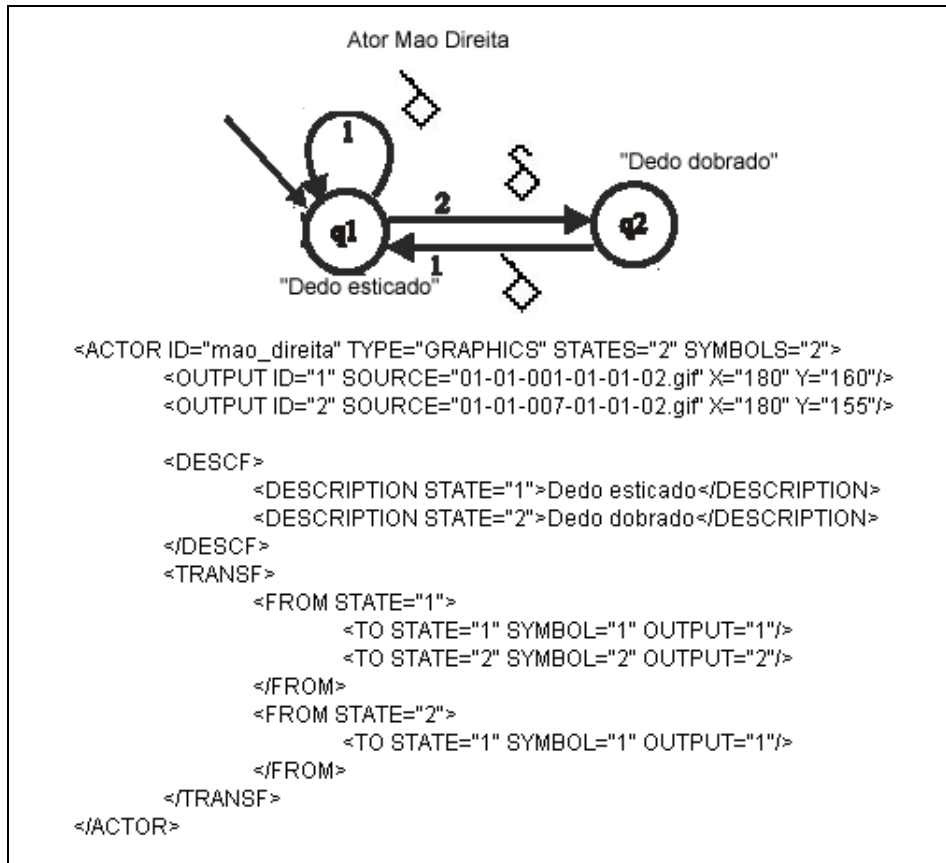


Figura 5.7: Exemplo do elemento *ACTOR* para a especificação do ator *mão direita*

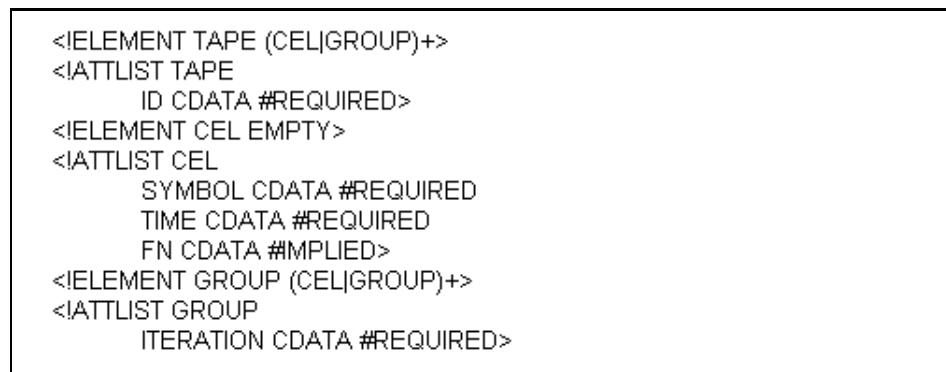


Figura 5.8: Trecho da DTD com os elementos *TAPE*, *CEL* e *GROUP*

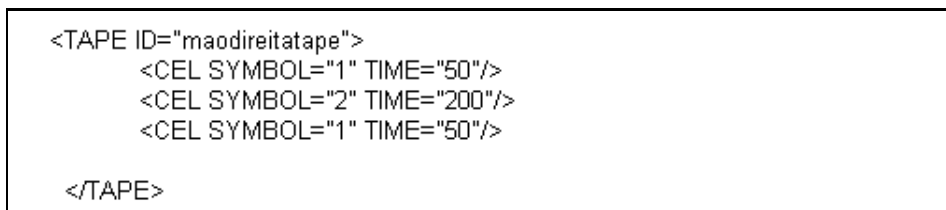


Figura 5.9: Exemplo do elemento *TAPE* para especificação da fita de entrada

ocorra. Por exemplo, até que o botão do mouse seja pressionado. O símbolo *EMPTY* indica que o ator deve permanecer no estado atual e as transformações descritas em *FN* devem ser aplicadas à imagem de saída mais recente. A utilização do *EMPTY* é con-

veniente para evitar a criação de arestas na forma de laço nos estados do ator só com o objetivo de aplicar transformações, o que provoca um aumento na representação da função de transição.

### 5.2.5 Especificação das Instâncias do Ator AGA

A associação entre a especificação do ator AGA e uma fita de entrada é realizada pelo elemento *INSTANCE* (Figura 5.10). Através deste elemento é possível criar várias instâncias do mesmo ator, assim como ter a mesma fita de entrada compartilhada por vários atores.

```
<ELEMENT INSTANCE (USE+)>
<!ATTLIST INSTANCE
  ID CDATA #IMPLIED
  ACTOR CDATA #REQUIRED
  ORDER CDATA #IMPLIED>
<ELEMENT USE EMPTY>
<!ATTLIST USE
  TAPE CDATA #REQUIRED>
```

Figura 5.10: Trecho da DTD com o elemento *INSTANCE* e *USE*

O atributo *ACTOR* determina qual a especificação de ator utilizada, enquanto os elementos filhos *USE* determinam quais são as fitas de entrada (*TAPE*) que devem ser lidas. As fitas de entrada são utilizadas uma após a outra durante a apresentação da animação. O atributo *ORDER* indica qual a camada de atuação da instância (relação de ordem definida no modelo AGA). O atributo *ID* de *INSTANCE* determina uma identificação para cada instância. Esta identificação é importante para suportar a construção de predicados destinados a consultas baseadas nos estados dos atores pelo AGA Player. Por exemplo, através da análise da fita andar (Figura 5.9) e o autômato do ator bicho (Figura 5.7) é possível recuperar o momento exato em que a instância bicho faminto (Figura 5.11) ocupa um estado cuja descrição é "alerta". A figura 4.10 ilustra a criação de duas instâncias, bicho faminto e bicho lento, a partir da mesma especificação de ator bicho. A primeira é associada à fita de entrada andar enquanto a segunda instância associa a outra fita andar lento.

```
<INSTANCE ID="maodireita" ACTOR="mao_direita" ORDER="2">
  <USE TAPE="maodireitatape"/>
</INSTANCE>
```

Figura 5.11: Exemplo do elemento *INSTANCE* para a criação de instâncias

## 5.3 AGA Player

O programa de visualização, AGA Player, foi desenvolvido em linguagem Java na forma de applet. A escolha pelo Java foi motivada principalmente pela necessidade de portabilidade do visualizador e o amplo repositório de objetos existentes para manipulação gráfica e processamento do XML encontrado na linguagem.

Como descrito anteriormente, o AGA Player é executado no cliente para realizar a reprodução da animação com base nas especificações em AgaML. A execução do AGA

Player pode ser dividida esquematicamente em três fases:

1. **Carga do AgaML e mídias externas** - O applet, ao ser inicializado, carrega o arquivo em AgaML cujo nome foi passado como parâmetro. Uma vez carregado, o arquivo AgaML é processado, e as informações das especificações dos atores e fitas são extraídas. Entre as informações estão as referências para os arquivos de imagens e sons utilizados, os quais são carregados na seqüência, e armazenados em estruturas de dados do applet. Como as mídias se encontram em arquivos independentes, o processo de carga aguarda a confirmação de leitura de todas antes de iniciar as outras fases;
2. **Construção das instâncias** - A partir das especificações das instâncias extraídas do AgaML, os atores AGA são efetivamente instanciados em estruturas de dados internas e associados às suas respectivas fitas de entrada. Após esta fase, as instâncias se encontram no estado inicial e prontas para serem simuladas;
3. **Simulação dos autômatos e geração dos quadros** - Os autômatos devem ser simulados em paralelo para obter a formação dos quadros de saída. Porém, a implementação do modelo utilizando processos concorrentes para cada autômato (multithreading) implica em uma dificuldade adicional para a sincronização precisa das trocas de imagens dos autômatos. Assim, uma alternativa de implementação foi escolhida para garantir que as imagens geradas representem fielmente as especificações determinadas pelas fitas de entrada. Um processo de temporização (timer) foi criado para gerar eventos de análise dos autômatos com a frequência determinada pelo *FRAMERATE*. A cada evento do timer, as transições pertinentes são realizadas em cada um dos autômatos e o quadro de saída é formado, reproduzindo assim a imagem respectiva para aquele instante na animação. Durante o processo de simulação, o AGA Player está disponível para intervenções de reprodução e consulta solicitadas pelo observador através dos botões de controle.

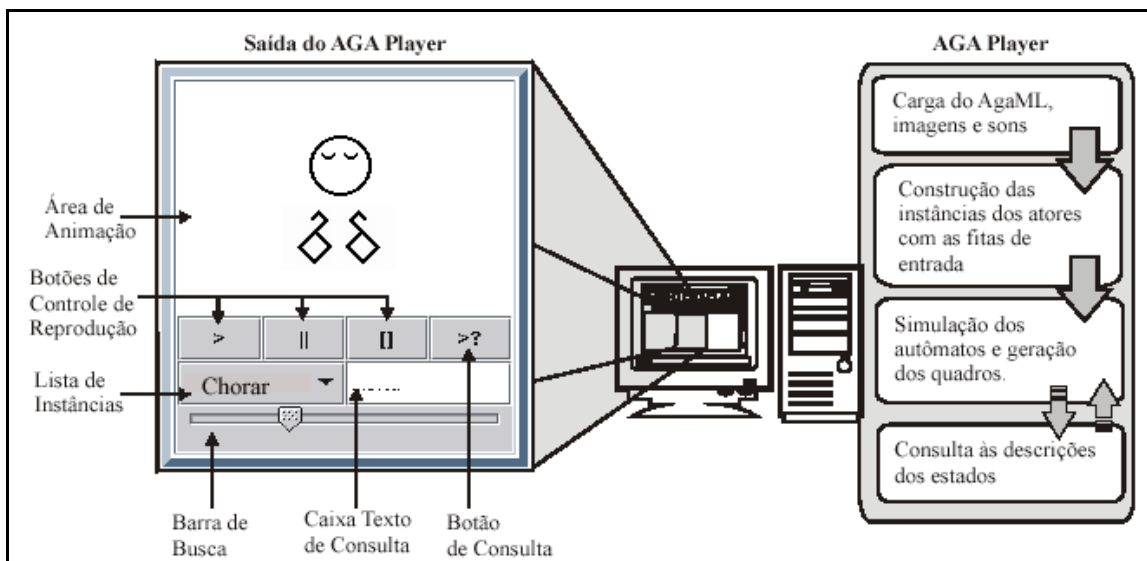


Figura 5.12: Painel de controle do AGA Player visualizado pelo observador

A Figura 5.12 ilustra o painel de controle do AGA Player visualizado pelo observador durante visita à página Web. O painel conta com 3 botões de controle de reprodução



para as funções corriqueiras de reproduzir, pausar e parar. Uma barra de busca é disponibilizada para o usuário com o objetivo de indicar o tempo decorrido de animação e proporcionar a busca temporal da seqüência animada.

O AGA Player conta com um recurso de busca incomum nos programas de visualização disponíveis na Internet. Este recurso é particular do modelo AGA e proporciona um mecanismo refinado de pesquisa baseado em palavras-chave. A lista de instâncias, a caixa de texto e o botão de consulta dão suporte operacional para a execução deste mecanismo.

O procedimento de pesquisa é operacionalmente simples. O observador escolhe um ator na lista de instâncias e digita alguma palavra-chave que represente a situação semântica que deseja verificar. Assim, quando o botão de consulta é pressionado, é realizada uma análise na instância, investigando se em algum instante da animação ela ocupa um estado cuja descrição vinculada coincide com a palavra-chave requerida. Caso afirmativo, a animação é posicionada neste instante, pronta para iniciar a reprodução. A análise é realizada a partir do instante corrente. Desta maneira, o botão pode ser pressionado sucessivas vezes a fim de pesquisar novas situações ao longo da animação. A potencialidade de casamento das palavras-chave cresce à medida que o projetista da animação associa um amplo vocabulário aos estados através da função descrição.

## 6 TRADUTOR SWML/AGAML

Este capítulo está dividido em duas seções. A primeira trata de uma etapa importante do trabalho, que foi o estudo detalhado dos símbolos *SW* a fim de conhecer as suas características e seus significados. A segunda trata do Tradutor de Textos SWML/AgAML, responsável por fazer a tradução de um sinal estático em animação.

### 6.1 Estudo de Textos *SW*

Os símbolos *SW* estão classificados segundo suas categorias e seguem uma ordem específica, denominada SSS, Sign-Symbol-Sequence (Tabela 6.1) (Capítulo 2) e representado por

$$SSS = (C, G, S, V, F, R)$$

onde:

$C \in \{1, \dots, c\}$ , sendo que cada elemento é uma *categoria* de símbolos.

$G \in \{1, \dots, g\}$ , sendo que cada elemento é um *grupo* em uma *categoria*  $C$ .

$S \in \{1, \dots, s\}$ , sendo que cada elemento é um *símbolo* de um *grupo* de  $C$ .

$V \in \{1, \dots, v\}$ , sendo que cada elemento representa a *variação* de um *símbolo* de um *grupo* de  $C$ .

$F \in \{1, \dots, f\}$ , sendo que cada elemento representa um *preenchimento* de um *símbolo* de um *grupo* de  $C$ .

$R \in \{1, \dots, r\}$ , sendo que cada elemento representa a *rotação* de um *símbolo* de um *grupo* de  $C$ .

Tabela 6.1: SSS de um símbolo *SW*

| Símbolo | Categoria | Grupo | N_Símbolo | Variação | Preenchimento | Rotação |
|---------|-----------|-------|-----------|----------|---------------|---------|
| *       | 02        | 01    | 001       | 01       | 01            | 01      |

Os textos escritos em *SW* são armazenados em arquivos de sinais. Para visualizar os SSS dos símbolos que compõem um sinal, usa-se o conversor de textos *SW* para textos *SWML* (Capítulo 2). O documento *SWML* lista todas as informações a respeito do sinal escrito em *SW*, tornando possível a análise dos símbolos que fazem parte dele. A partir do conhecimento dos símbolos e suas características, principalmente dos movimentos, é possível determinar algumas regras que serão responsáveis pela animação dos sinais.

A partir de sinais extraídos da LIBRAS, e escritos através do SWEdit, foram criadas algumas animações<sup>1</sup>. O objetivo foi avaliar as melhores possibilidades de como apresentá-las, que partes do corpo e que elementos (símbolos) seriam usados.

Como visto no Capítulo 2, os símbolos SW (SSS-2004) estão divididos em 8 categorias e seus elementos são descritos como nas Tabelas 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1.

Tabela 6.2: Categoria 1 - Mãos

| Grupo   | Configuração da Mão              | Símbolos |
|---------|----------------------------------|----------|
| 01 - 01 | Dedo Indicador                   |          |
| 01 - 02 | Dedos Médio e Indicador          |          |
| 01 - 03 | Dedos Médio, Indicador e Polegar |          |
| 01 - 04 | Quatro Dedos                     |          |
| 01 - 05 | Cinco Dedos                      |          |
| 01 - 06 | Dedo Mínimo                      |          |
| 01 - 07 | Dedo Anelar                      |          |
| 01 - 08 | Dedo Médio                       |          |
| 01 - 09 | Dedos Polegar e Indicativo       |          |
| 01 - 10 | Dedo Polegar                     |          |

Tabela 6.3: Categoria 2 - Movimentos

| Grupo   | Tipo de Movimento                         | Símbolos    |
|---------|---|-------------|
| 02 - 01 | Contato                                   | * + # @     |
| 02 - 02 | Movimento dos Dedos                       | • ◦ ∨ ^ ~ ≡ |
| 02 - 03 | Reto paralelo ao plano da parede          | ↑ ↑↓ ↑↗ ↖   |
| 02 - 04 | Reto paralelo diagonal ao plano da parede | ↗ ↘ ↙ ↚     |
| 02 - 05 | Reto Paralelo ao plano do assoalho        | ↑ ↗ ↓ ↘ ↖ ↙ |
| 02 - 06 | Curva paralela ao plano da parede         | ↪ ↻ ↺ ↻ ↺   |
| 02 - 07 | Curva “batendo” no plano da parede        | ↻ ↺ ↻ ↺ ↻ ↺ |
| 02 - 08 | Curva “batendo” no plano do assoalho      | ↻ ↺ ↻ ↺ ↻ ↺ |
| 02 - 09 | Curva paralela ao plano do assoalho       | ↪ ↻ ↺ ↻ ↺   |
| 02 - 10 | Circular                                  | ⊙ ⊙ ⊙       |

<sup>1</sup><http://atlas.ucpel.tche.br/~rubiad/tabela2.htm>

Tabela 6.4: Categoria 3 - Expressões Faciais

| Grupo   | Configuração da Face   | Símbolos |
|---------|------------------------|----------|
| 03 - 01 | Testa                  |          |
| 03 - 02 | Olhos                  |          |
| 03 - 03 | Olhar                  |          |
| 03 - 04 | Orelhas                |          |
| 03 - 05 | Nariz                  |          |
| 03 - 06 | Boca                   |          |
| 03 - 07 | Língua                 |          |
| 03 - 08 | Dentes                 |          |
| 03 - 09 | Queixo                 |          |
| 03 - 10 | Pescoço-Cabelo-Emoções |          |

Tabela 6.5: Categoria 4 - Cabeça

| Grupo   | Configuração da Cabeça                     | Símbolo |
|---------|--|---------|
| 04 - 01 | Pontos de Vista da Cabeça                  |         |
| 04 - 02 | Bordas da Cabeça                           |         |
| 04 - 03 | Cabeça se move paralelamente ao plano      |         |
| 04 - 04 | Cabeça se move perpendicularmente ao plano |         |
| 04 - 05 | Posições da Cabeça                         |         |

Tabela 6.6: Categoria 5 - Tronco

| Grupo   | Configuração do Tronco                      | Símbolo |
|---------|---|---------|
| 05 - 01 | Pontos de Vista dos Ombros                  |         |
| 05 - 02 | Posição dos Ombros                          |         |
| 05 - 03 | Ombros se movem paralelamente ao plano      |         |
| 05 - 04 | Ombros se movem perpendicularmente ao plano |         |
| 05 - 05 | Movimento do Torso                          |         |

Tabela 6.7: Categoria 6 - Membro



| Grupo   | Configuração do Membro | Símbolo  |
|---------|------------------------|--|
| 06 - 01 | Braço-Cotovelo-Pulso   |  |
| 06 - 02 | Movimento do Corpo     |  |

Tabela 6.8: Categoria 7 - Localização

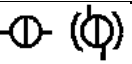




| Grupo   | Localização  | Símbolo  |
|---------|--------------|--|
| 07 - 01 | Corpo        |  |
| 07 - 02 | Profundidade |  |
| 07 - 03 | Espaço       |  |

Tabela 6.9: Categoria 8 - Dinâmicas e Pontuação

| Grupo   | Dinâmicas                | Símbolo   |
|---------|--------------------------|---|
| 08 - 01 | Qualidade do Movimento   |  |
| 08 - 02 | Sincronismo do Movimento |  |
| 08 - 03 | Pontuação de Sinal       |  |
| 08 - 04 | Marcação de Movimentos   |  |

### 6.1.1 Estudo dos Movimentos

Como citado anteriormente, o comportamento dos símbolos de um sinal a serem animados são definidos conforme o movimento descrito no texto *SW*. Esses movimentos podem ser retilíneos, curvos, circulares, em zig-zag, de cruzamento, de alternância e de alteração de rotação. Cada movimento pode ter até oito rotações e até quatro variações. O deslocamento é determinado pela combinação da rotação e da variação, isto é, a rotação indica a direção do movimento e a variação indica a distância a ser percorrida pelo símbolo em movimento a partir de um estado inicial.

Partindo das animações geradas, descreve-se alguns dos símbolos de movimentos disponíveis no sistema *SW* que foram estudados para o desenvolvimento deste trabalho.

#### 6.1.1.1 Contatos

Os símbolos desse grupo são os que representam contato entre as mãos ou entre as mãos e a face ou entre as mãos e outras partes do corpo. Eles pertencem ao Grupo 1 da

Categoria 2. A seguir exemplos de alguns dos símbolos analisados (Tabela 6.1.1.1).

Tabela 6.10: SSS dos movimentos de contato analisados

| Símbolo | Categoria | Grupo | Numero Símbolo | Variação | Preenchimento | Rotação |
|---------|-----------|-------|----------------|----------|---------------|---------|
| *       | 02        | 01    | 001            | 01       | 01            | 01      |
| #       | 02        | 01    | 005            | 01       | 01            | 01      |
| ⊙       | 02        | 01    | 007            | 01       | 01            | 01      |
| @       | 02        | 01    | 009            | 01       | 01            | 01      |

Na Figura 6.1, as mãos se tocam rápida e gentilmente, e é representado pelo símbolo Contato (*touch*), de número 001, conforme Tabela 6.1.1.1.

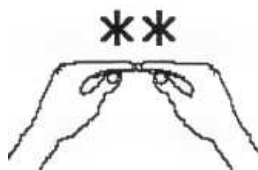


Figura 6.1: Representação de um Contato duplo (COSTA, 2006b)

Na Figura 6.2, o símbolo de número 005 representa o contato Bater (*strike*), definido com uma mão contatando fortemente uma determinada superfície.



Figura 6.2: Contato Bater - sinal “Pagar” (COSTA, 2006b)

Na Figura 6.3, o símbolo de número 007 representa o contato Escovar (*brush*), definido como um movimento que contata uma determinada superfície e se move sobre ela, mas não permanece.

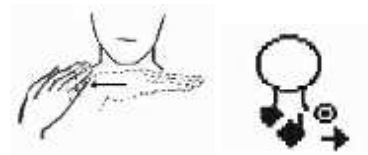


Figura 6.3: Contato Escovar - sinal “Morrer” (COSTA, 2006b)

Na Figura 6.4, o símbolo de número 009 representa o contato Esfregar (*rub*), definido como um contato que se move sobre uma determinada superfície e permanece sobre ela.

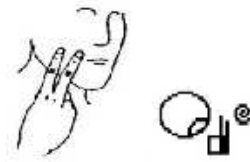


Figura 6.4: Contato Esfregar - sinal “Vinho” (COSTA, 2006b)

### 6.1.1.2 Movimentos Verticais

As setas duplas representam um movimento vertical (palma da mão paralela ao plano da parede). Nos movimentos verticais retilíneos, o objeto poderá se deslocar para cima ou para baixo, ou nas diagonais para cima e para baixo (Figura 6.5). O movimento é plano na frente do corpo e pertence ao Grupo 3 da Categoria 2.

Tabela 6.11: SSS do Movimento Vertical

| Símbolo | Categoria | Grupo | Numero Símbolo | Variação  | Preenchimento | Rotação |
|---------|-----------|-------|----------------|-----------|---------------|---------|
| ↑↑      | 02        | 03    | 001            | <b>01</b> | 01            | 01      |
| ↑↑↑     | 02        | 03    | 001            | <b>02</b> | 01            | 01      |
| ↑↑↑↑    | 02        | 03    | 001            | <b>03</b> | 01            | 01      |
| ↑↑↑     | 02        | 03    | 002            | 01        | 01            | 01      |
| ↓↑      | 02        | 03    | 003            | 01        | 01            | 01      |
| ↖↗      | 02        | 03    | 004            | 01        | 01            | 01      |
| ↑↑↑↑    | 02        | 03    | 005            | 01        | 01            | 01      |
| ↑↓↑     | 02        | 03    | 006            | 01        | 01            | 01      |

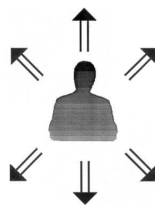


Figura 6.5: Representação de movimentos verticais (COSTA, 2006b)

O símbolo 001, pertencente ao grupo 3 da Categoria 2, representa um deslocamento retilíneo para cima ou para baixo, como na Figura 6.6. Ele pode representar até três distâncias percorridas diferentes: curta, média e longa. Isso depende de sua variação no SSS. Veja Tabela 6.1.1.2.

O símbolo 002 representa um duplo deslocamento curto, isto é, a mão irá fazer duas vezes o mesmo movimento, assim como o símbolo 005 irá fazer três vezes o mesmo movimento. Já o símbolo 003 faz o movimento alternado, para cima e para baixo. Assim

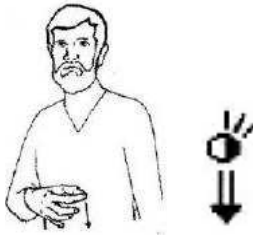


Figura 6.6: Movimento vertical - sinal “Exato” (COSTA, 2006b)

como o símbolo 006 alterna mais um movimento, para cima e para baixo e para cima de novo. O símbolo 004 representa um movimento cruzado, como se a mão fizesse um “sinal da cruz”.

Os símbolos dessa categoria possuem três formas de preenchimento (*fill*) e até 8 rotações. O preenchimento informa a mão que fará parte do movimento, direita ou esquerda, ou as duas unidas em um só movimento.

### 6.1.1.3 Movimentos Horizontais

As setas simples com as pontas fechadas representam um movimento horizontal (palma da mão paralela ao chão) e pertencem ao Grupo 5 da Categoria 2. O objeto poderá se mover para frente ou para trás, ou nas diagonais para frente ou para trás.

Tabela 6.12: SSS do Movimento Horizontal

| Símbolo | Categoria | Grupo | Número Símbolo | Variação  | Preenchimento | Rotação |
|---------|-----------|-------|----------------|-----------|---------------|---------|
| ↑       | 02        | 05    | 001            | <b>01</b> | 01            | 01      |
| ↑       | 02        | 05    | 001            | <b>02</b> | 01            | 01      |
| ↑       | 02        | 05    | 001            | <b>03</b> | 01            | 01      |
| ↑↑      | 02        | 05    | 002            | 01        | 01            | 01      |
| ↓↑      | 02        | 05    | 003            | 01        | 01            | 01      |
| ←↑      | 02        | 05    | 004            | 01        | 01            | 01      |
| ↑↑↑     | 02        | 05    | 005            | 01        | 01            | 01      |
| ↑↓↑     | 02        | 05    | 006            | 01        | 01            | 01      |

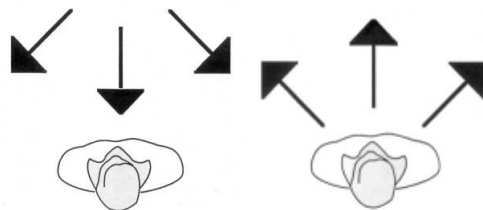


Figura 6.7: Representação de movimentos horizontais para trás e para frente (COSTA, 2006b)



O símbolo 001, pertencente ao grupo 5 da Categoria 2, representa um deslocamento retilíneo para frente ou para trás, como na Figura 6.8. Assim como no movimento vertical, a variação pode representar até três distâncias percorridas diferentes: curta, média e longa. Veja Tabela 6.1.1.3.

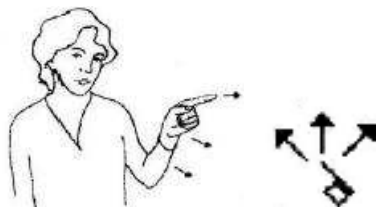




Figura 6.8: Movimento horizontal - sinal *Eles* (COSTA, 2006b)

O símbolo 002 representa um duplo deslocamento curto, isto é, a mão irá fazer duas vezes o mesmo movimento, duas vezes para cima ou duas vezes para baixo, assim como o símbolo 005 irá fazer três vezes o mesmo movimento. Já o símbolo 003 faz o movimento alternado, para frente e para trás. Assim como o símbolo 006 alterna mais um movimento, para frente e para trás e para frente de novo. O símbolo 004 representa um movimento cruzado, como se a mão fizesse um “sinal da cruz” no plano paralelo ao chão.

#### 6.1.1.4 Movimentos Circulares

Os movimentos circulares são do Grupo 10 da Categoria 2, como apresentados na Tabela 6.1.1.4. Os movimentos analisados podem ter até oito rotações, dependendo da direção e do ponto inicial do movimento.

Tabela 6.13: SSS dos Movimentos Circulares

| Símbolo   | Categoria | Grupo | Numero Símbolo | Varição | Preenchimento | Rotação |
|---|-----------|-------|----------------|---------|---------------|---------|
|  | 02        | 10    | 001            | 01      | 01            | 01      |
|  | 02        | 10    | 001            | 01      | 01            | 01      |

Os símbolos da Figura 6.9 representam movimentos circulares feitos pelo cotovelo. Este símbolo pode apontar para oito direções diferentes, como ilustrado no exemplo da Figura 6.10.



Figura 6.9: Movimentos Circulares para cima e para baixo (COSTA, 2006b)

No exemplo da Figura 6.10, as mãos fazem um movimento simultâneo circular que inicia nas laterais de baixo para cima. A mão esquerda é movimentada da direita para a esquerda, de baixo para cima. Simultaneamente, a mão direita é movimentada da esquerda para a direita, de baixo para cima. As setas duplas indicam que o movimento se repetirá.



Figura 6.10: Exemplo de movimento circular, sinal "Computador"

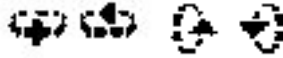


Figura 6.11: Movimentos circulares para frente e para trás e movimentos circulares para cima e para frente e para baixo e para trás (COSTA, 2006b)

Os símbolos da Figura 6.11 representam o movimento circular em outras direções.

No exemplo da Figura 6.12, o movimento é feito da frente para trás, duas vezes. A repetição do movimento está representada pelas setas duplas do símbolo de movimento.



Figura 6.12: Exemplo de movimento circular para trás, sinal "Sozinho"



Figura 6.13: Exemplo de movimento circular para trás, sinal "Brincar"

No exemplo da Figura 6.13, o movimento das mãos é simultâneo e circular para baixo e para trás, duas vezes.

### 6.1.2 Resultados do Estudo de Textos SWML

Através do estudo dos textos SWML, como citado no início do capítulo, tornou-se possível o conhecimento detalhado dos símbolos, suas características e seus significados. A partir disso foram definidas as regras de movimentos, que são responsáveis pelo comportamento dos símbolos na animação.

No exemplo da Figura 6.14, o elemento *category* apresenta o tipo de símbolo, nesse caso um *movimento*. O elemento *group* identifica o tipo de movimento como *vertical*. O elemento *symbnum* identifica o número do símbolo dentro do grupo e vem seguido do elemento *variation* que identifica se o movimento vertical é curto, médio ou longo. O elemento *fill* identifica se o movimento é realizado pela mão esquerda ou direita. Por fim, o elemento *rotation* identifica a direção do movimento, se o símbolo se deslocará para a esquerda ou direita, para cima ou para baixo, para a diagonal para cima ou diagonal para baixo.

```

<symb x="61" y="90" x-flop="1" y-flop="0" color="0,0,0">
  <category>02</category>
  <group>03</group>
  <symbnum>001</symbnum>
  <variation>01</variation>
  <fill>01</fill>
  <rotation>01</rotation>

  Analysis
    Category: 02 - Movement
    Grupo: 03 Straight Wall Plane
    Num_Symbol: 001
    Variation: 01 Straight Wall Plane Small
    Fill: 01 Right Hand
    Rotation: 01 Righth Hand Up

</symb>

```

Figura 6.14: Exemplo de um trecho de texto analisado

Como dito anteriormente, os símbolos que representam os movimentos são responsáveis pelas mudanças ocorridas durante a animação, por isso as regras são determinadas a partir de suas características.

Identificado o tipo de movimento escrito através do SSS, as regras são determinadas levando-se em conta as partes do corpo envolvidas no sinal. Por exemplo, um texto escrito em *SW* que tem um símbolo que representa um movimento vertical para cima e um símbolo que representa a cabeça com borda no queixo representa o gesto o qual a mão fará o movimento em direção ao queixo (Figura 6.15).

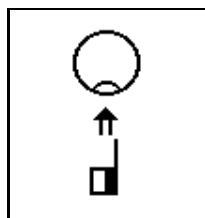


Figura 6.15: Exemplo de um gesto escrito em *SW* com movimento vertical

É comum acontecer movimentos combinados, como por exemplo contato e movimento vertical (Figura 6.16). Assim como no exemplo da Figura 6.15, o movimento da mão será em direção ao queixo e acontecerá um contato leve. Quanto mais informações forem fornecidas durante a edição dos textos, mais precisa será a animação dos gestos.

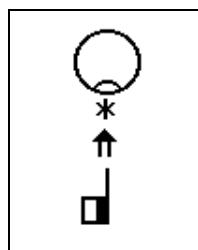


Figura 6.16: Exemplo de um gesto escrito em *SW* com movimentos combinados

Nas subseções a seguir serão apresentados exemplos de alguns movimentos e suas respectivas regras de movimentação para os atores constituintes dos sinais.

### 6.1.3 Definindo Regras de Transições de Estados para Movimentos Verticais

Os símbolos que representam esse movimento são responsáveis pelo deslocamento das mãos para cima e para baixo e diagonais para cima e para baixo como na Figura 6.17.

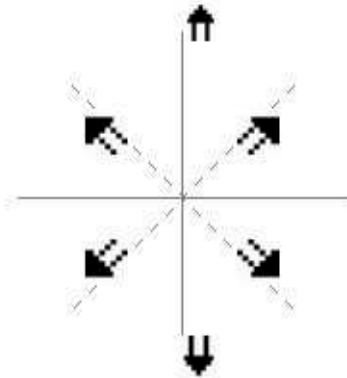


Figura 6.17: Direções dos movimentos verticais

Esses símbolos podem representar três (3) deslocamentos distintos (curto, médio e longo), dependendo de suas variações. E a direção do movimento é dada pela rotação do símbolo.

O sinal apresentado na Figura 6.18, por exemplo, tem um deslocamento curto das mãos, representado pelo símbolo que representa o movimento retilíneo vertical para baixo com variação 1.

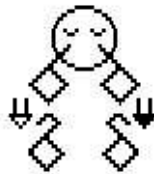


Figura 6.18: Escrita do Sinal *Chorar*

O deslocamento das mãos neste exemplo está representado na Figura 6.19, onde  $P_i$  é a posição inicial das mãos e  $P_f$  a posição final.

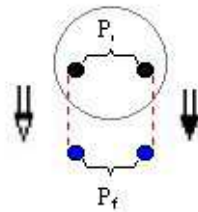


Figura 6.19: Exemplo de deslocamento para o sinal *Chorar*

Então, a partir da variação e da rotação do símbolo é possível saber a direção e o

deslocamento do objeto durante a animação do sinal. A regra que define este movimento está representada na Figura 6.20.

```

Se (movimento = vertical) então
  Se (variação = movimento curto) então
    d <- deslocamento equivalente ao movimento curto
    Se (rotação = seta para baixo) então
      deslocamento do objeto <- (xi, yi - d)
    Fim Se
  Fim Se
Fim Se

```

Figura 6.20: Exemplo algorítmico para regras de movimentos verticais

O deslocamento do objeto se dá através da posição inicial do objeto ( $x_i, y_i$ ) acrescida ou diminuída da distância equivalente ao movimento (curto, médio ou longo). No exemplo acima, o objeto se desloca para baixo, sobre o eixo das ordenadas ( $y$ ), não alterando a posição sobre o eixo das abcissas ( $x$ ). Para este tipo de movimento o objeto se deslocará do ponto inicial ao ponto final, sendo interpolado por apenas dois pontos (inicial e final como mostrado na Figura 6.19).

#### 6.1.4 Definindo Regras de Transição de Estados para Movimentos Horizontais

Os símbolos que representam o movimento horizontal são responsáveis pelo deslocamento das mãos para frente e para trás e para as diagonais para frente e para trás como na Figura 6.21.

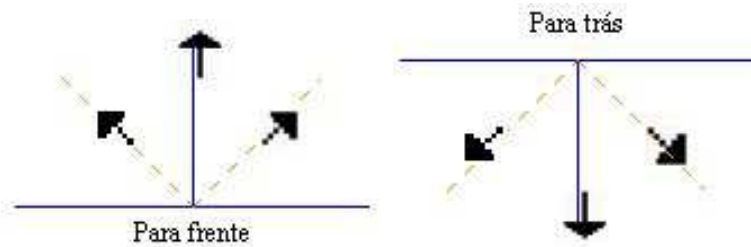


Figura 6.21: Direções dos movimentos horizontais

Assim como nos movimentos verticais, os movimentos horizontais também podem ter três (3) deslocamentos distintos: curto, médio e longo.

O sinal apresentado na Figura 6.22, por exemplo, tem um deslocamento curto das mãos, representado pelo símbolo que representa o movimento retilíneo horizontal diagonal para a frente com variação 1.



Figura 6.22: Escrita do Sinal *Idéia*

O deslocamento das mãos neste exemplo está representado na Figura 6.23, onde  $P_i$  é o posição inicial das mãos e  $P_f$  a posição final. Para demonstrar o movimento horizontal diagonal para frente aumenta-se o tamanho do símbolo, no caso a mão, para se ter a impressão de movimento para a frente.

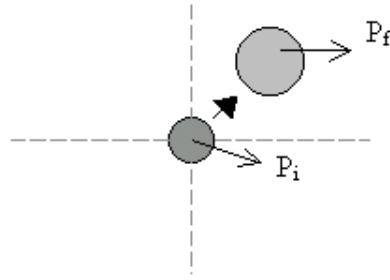


Figura 6.23: Exemplo do deslocamento do Sinal *Idéia*

Então, a partir da variação e da rotação do símbolo é possível saber a direção e o deslocamento do objeto durante a animação do sinal. A regra que define este movimento está apresentada na Figura 6.24.

Em casos de movimentos horizontais diagonais e laterais, o deslocamento do objeto se dá através da posição inicial do objeto  $(x_i, y_i)$  acrescida ou diminuída da distância equivalente ao movimento (curto, médio ou longo) juntamente com o aumento (para frente) ou diminuição (para trás) do objeto, sendo interpolado por apenas dois pontos (inicial e final). O objeto terá seu tamanho aumentado conforme o seu deslocamento (variação). No exemplo acima, o objeto se desloca na diagonal direita para frente.

```
Se (movimento = horizontal) então
  Se (variação = movimento curto) então
    d <- deslocamento equivalente ao movimento curto
    Se (rotação = seta diagonal para frente) então
      deslocamento do objeto <- (xi + d, yi + d)
      objeto <- objeto ampliado
    Fim Se
  Fim Se
Fim Se
```

Figura 6.24: Exemplo algorítmico para regras de movimentos horizontais

Em casos de movimentos horizontais para frente, o deslocamento será representado pelo aumento do objeto e, em movimentos para trás, o deslocamento será representado pela diminuição do objeto durante a animação. O objeto não faz nenhuma movimentação, apenas é aumentado ou diminuído de tamanho.

### 6.1.5 Definindo Regras de Transição de Estados para Movimentos Circulares

O sinal da Figura 6.25, apresenta um movimento circular que começa de baixo para cima e que irá se repetir, como indicado pelas setas.



Figura 6.25: Escrita do Sinal *"Importante"*

O deslocamento das mãos, neste exemplo, está representado na Figura 6.26, onde  $P_1$  é o posição inicial das mãos e  $P_5$  a posição final. O movimento circular é representado por uma seqüência de cinco (5) posições.

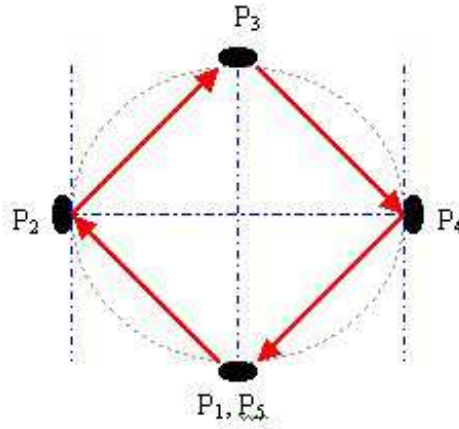


Figura 6.26: Deslocamentos das mãos para o Sinal "Importante"

Então, neste caso, a partir da rotação do símbolo é possível saber a direção do movimento e se há repetição do movimento durante a animação do sinal. A regra que define este movimento está apresentada na Figura 6.27.

```

Se (movimento = circular para cima ou para baixo) então
  Se (rotação = seta dupla indicando movimento de baixo para cima para a esquerda) então
    repetir <- 0
    Repita
      deslocamento <- deslocamento equivalente ao movimento circular
      P1 <- (x1 , y1)
      P2 <- x1 - deslocamento, y1 + deslocamento)
      P3 <- (x2 + deslocamento, y2 + deslocamento)
      P4 <- (x3 + deslocamento, y3 - deslocamento)
      P5 <- (x4 - deslocamento, y4 - deslocamento)
      Animação <- Sequência(P1 , P2 , P3 , P4 , P5 )
      repetir <- repetir + 1
    Até (repetir = 2)
  Fim Se
Fim Se

```

Figura 6.27: Exemplo algorítmico para regras para movimentos circulares para cima e para baixo

Em casos de movimentos circulares são definidas cinco (5) posições, sendo que a posição final ocupa o mesmo ponto da posição inicial. Esse tipo de movimento é interpolado por cinco pontos e podem ser para a esquerda ou para direita, começar de baixo para cima ou de cima para baixo (Figura 6.9). Os movimentos também podem ser repetidos, como no exemplo da Figura 6.10.

Outro exemplo de movimento circular é o representado no exemplo da Figura 6.28, que apresenta um movimento para frente e para trás.



Figura 6.28: Escrita do Sinal "Sozinho"

O deslocamento das mãos neste exemplo está representado na Figura 6.29, onde  $P_1$  é o posição inicial das mãos e  $P_5$  a posição final. Esse movimento circular também é interpolado por cinco pontos.

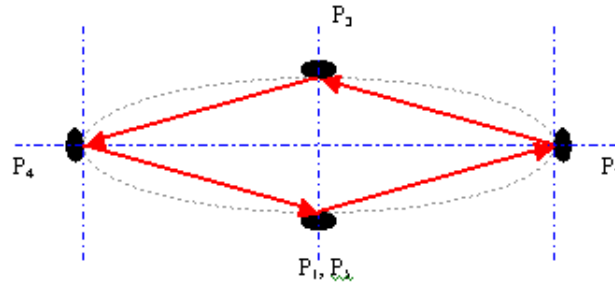


Figura 6.29: Deslocamento das mãos para o sinal "Sozinho"

Então, neste caso, a partir da rotação do símbolo é possível saber a direção do movimento e se há repetição do movimento durante a animação do sinal. A regra que define este movimento está apresentada na Figura 6.30.

```

Se (movimento = circular para frente e para trás) então
  Se (rotação = seta dupla indicando movimento de frente para trás para a direita) então
    repetir <- 0
    Repita
      deslocamento <- deslocamento equivalente ao movimento circular em helipce
      P1 <- (x1 , y1 )
      P2 <- (x1 - deslocamento, y1 + deslocamento)
      P3 <- (x2 + deslocamento, y2 + deslocamento)
      P4 <- (x3 + deslocamento, y3 - deslocamento)
      P5 <- (x4 - deslocamento, y4 - deslocamento)
      Animação <- Sequência(P1 , P2 , P3 , P4 , P5 )
      repetir <- repetir + 1
    Até (repetir = 2)
  Fim Se
Fim Se

```

Figura 6.30: Exemplo algorítmico para regras para movimentos circulares para frente e para trás

## 6.2 Tradutor de Textos SWML/AgamL

Esta seção apresenta o Tradutor de Textos SWML/AgamL (Figura 6.31).

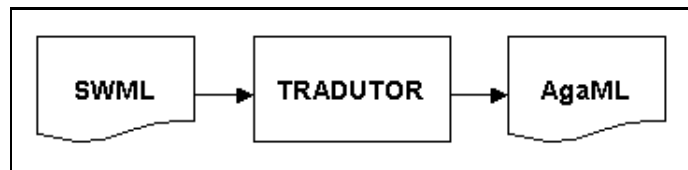


Figura 6.31: Entrada e saída de dados do Tradutor

Como citado anteriormente, os sinais escritos<sup>2</sup> em *SW* são salvos em SWML. Um

<sup>2</sup>Sinais escritos a partir dos editores que dispõem da notação *SW*, como o Sign Writer e o SW-Edit



documento SWML contém todas as informações (SSS) a respeito dos símbolos que compõem o sinal.

As regras, apresentadas na seção anterior, são implementadas no Tradutor para que a animação do sinal se comporte conforme o movimento representado no sinal escrito. Então, os SSS's que compõem o sinal são fornecidos ao Tradutor (Figura 6.32).

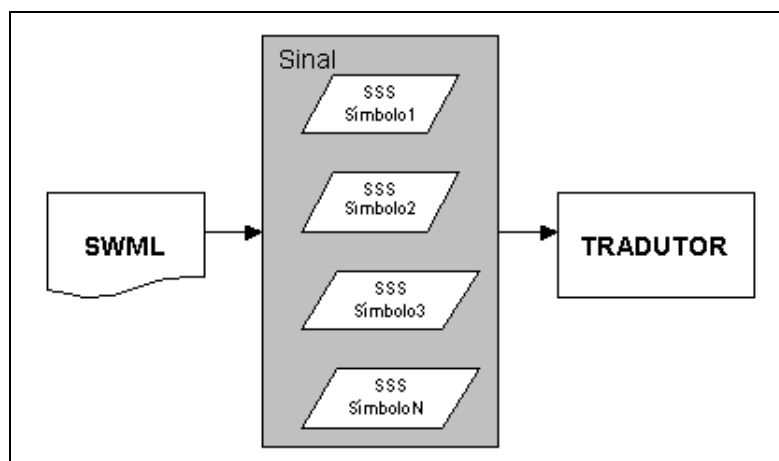


Figura 6.32: SSS de cada símbolo como entrada para o Tradutor

O processamento do Tradutor consiste em analisar os SSS's recebidos e montar um documento com as especificações da AgaML<sup>3</sup> conforme as características e comportamento de cada símbolo (Figura 6.33).

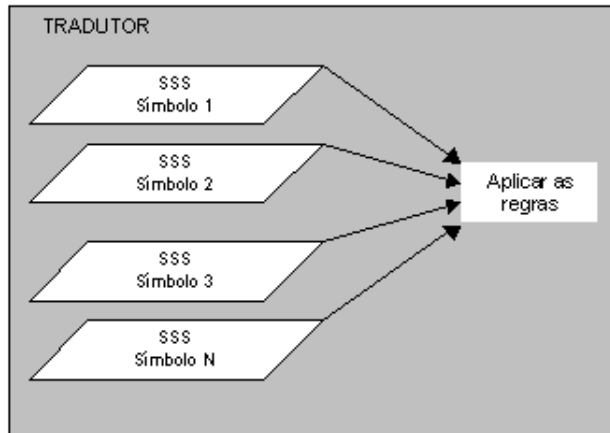


Figura 6.33: Processamento das informações

Aos SSS's (símbolos) que exprimem movimentos são aplicadas as regras de movimentos definidas na seção anterior. Eles determinam o movimento feito pelas mãos.

Os SSS's (símbolos) que não exprimem movimentos, como face e mãos, são transformados em atores SW. Cada SSS tem uma figura correspondente que representa um símbolo SW. Essas figuras são os atores SW e cada ator terá uma fita de entrada e uma (ou mais) instância(s). A fita de entrada reúne as informações que descrevem o comportamento do ator durante a animação e a instância faz a associação entre a especificação do ator e a fita de entrada (Capítulo 5). Então, esta estrutura compõe um documento AgaML com especificações para atores SW (Figura 6.34).

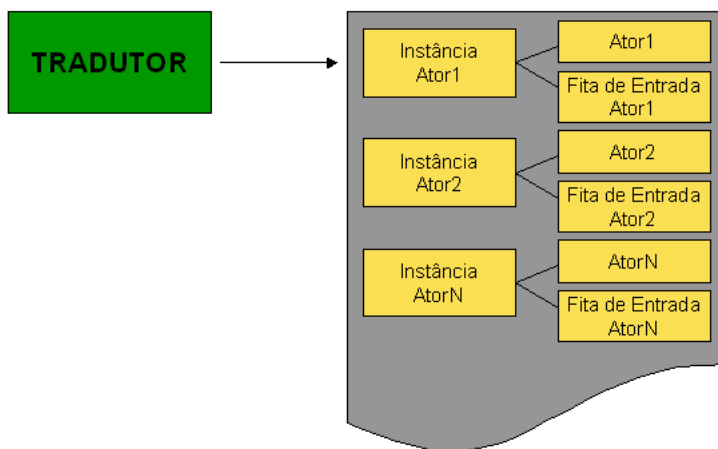


Figura 6.34: Dados de saída do Tradutor - documento AgaML



Figura 6.35: Sinal *Idéia*

A Figura 6.35 representa o sinal *Idéia* que é composto por quatro símbolos (face, mão direita, contato e movimento horizontal). Os SSS's desses símbolos são fornecidos para o Tradutor, que processa essas informações aplicando as regras de movimentos conforme os movimentos presentes na escrita do sinal (Figura 6.36).

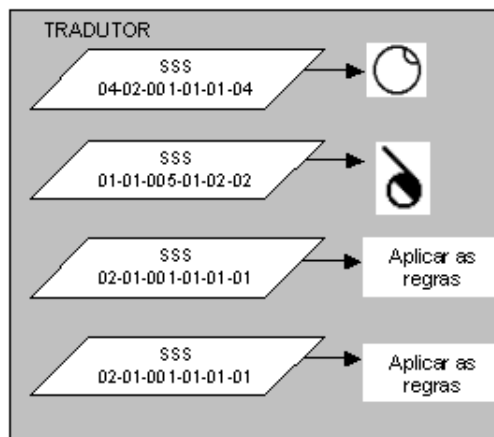


Figura 6.36: Processamento dos SSS's do sinal *Idéia*

O resultado desse processamento é um documento AgaML - com as especificações dos atores *face* e *mão direita* e suas respectivas fitas de entrada (Figura 6.37)- que servirá de arquivo de entrada para o AGA Player.

As Figuras 6.38 e 6.39 mostram, respectivamente, o trecho da estrutura dos documentos AgaML para os atores *cabeça* e *mão direita*, gerados pelo Tradutor. O elemento ACTOR contém elementos filhos que descrevem o alfabeto de saída  $\Delta$ (OUTPUT), a função descrição  $\sigma$  (DESCF) e a função transição temporal  $\delta_t$  (TRANSF). Os atributos ID,

<sup>3</sup>O AgaML é responsável pela especificação dos atores, fitas e instâncias AGA.

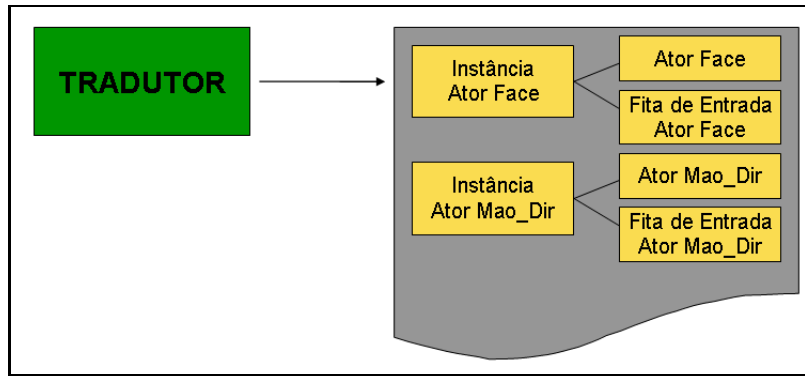


Figura 6.37: Resultado do processamento dos SSS's do sinal *Idéia*

TYPE, STATES e SYMBOLS dessa estrutura comportam, respectivamente, as informações sobre a identificação do ator, o seu tipo, as quantidades de estados e símbolos do seu alfabeto de entrada.

```
<ACTOR ID="ATOR1" TYPE="GRAPHICS" STATES="1" SYMBOLS="1">
  <OUTPUT ID="1" SOURCE=" 04-02-001-01-01-04.gif " X="36" Y="53"/>
  <DESCF>
    <DESCRIPTION STATE="1">CABEÇA</DESCRIPTION>
  </DESCF>
  <TRANSF>
    <FROM STATE="1">
      <TO STATE="1" SYMBOL="1" OUTPUT="1"/>
    </FROM>
  </TRANSF>
</ACTOR>
```

Figura 6.38: Elemento ACTOR para a especificação do ator *cabeça*

```
<ACTOR ID="ATOR2" TYPE="GRAPHICS" STATES="1" SYMBOLS="1">
  <OUTPUT ID="1" SOURCE=" 01-01-005-01-02-02.gif " X="75" Y="59"/>
  <DESCF>
    <DESCRIPTION STATE="1">Mão Direita</DESCRIPTION>
  </DESCF>
  <TRANSF>
    <FROM STATE="1">
      <TO STATE="1" SYMBOL="1" OUTPUT="1"/>
    </FROM>
  </TRANSF>
</ACTOR>
```

Figura 6.39: Elemento ACTOR para a especificação do ator *mão direita*

No alfabeto de saída (Figuras 6.38 e 6.39) o atributo *SOURCE* apresenta a imagem que representa o símbolo *SW*. O nome do arquivo da imagem é formado pelo SSS do símbolo.

A formação do nome do arquivo é feita pelo Tradutor, que concatena as informações retiradas do SWML do sinal (Figura 6.40) mostrando-as como no exemplo do ator *cabeça*, *SOURCE="04-02-001-01-01-04.gif"* (Figura 6.41). Todas as gifs que representam os símbolos *SW* estão neste formato de arquivo, representando seu SSS.

```
<symb x="36" y="53" x-flop="0" y-flop="0" color="0,0,0">
  <category>04</category>
  <group>02</group>
  <symbnum>001</symbnum>
  <variation>01</variation>
  <fill>01</fill>
  <rotation>04</rotation>
</symb>
```

Figura 6.40: Trecho do SWML com o SSS do símbolo da *cabeça*

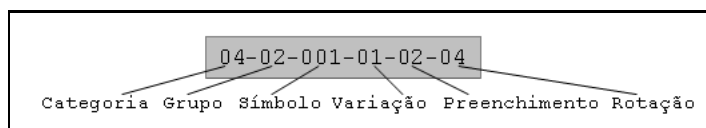


Figura 6.41: SSS do símbolo da *cabeça*

A fita de entrada reúne as informações que descrevem o comportamento do ator durante a animação. O elemento *TAPE* (Figura 6.42) contém uma coleção de elementos vazios *CEL* que descrevem cada célula da fita. Cada célula possui um símbolo de entrada (*SYMBOL*), um tempo de espera para o próximo símbolo (*TIME*) e a coleção de transformações que devem ser aplicadas à imagem de saída (*FN*). No exemplo da Figura 6.42 foi aplicado zoom ("SCALE 40") a imagem do ator para representar um movimento horizontal na diagonal para frente.

```
<TAPE ID="ATOR2TAPE">
  <CEL SYMBOL="1" TIME="200">
  <CEL SYMBOL="1" TIME="200" FN="SCALE 40;TRANSLATE 5 -30"/>
  <CEL SYMBOL="1" TIME="200">
</TAPE>
```

Figura 6.42: Elemento *TAPE* para a especificação de fita de entrada da *mão direita*

A associação entre a especificação do ator e a fita de entrada é realizada pelo elemento *INSTANCE*, como mostrado na Figura 6.43.

```
<INSTANCE ID="ATOR1IN" ACTOR="ATOR1" ORDER="1">
  <USE TAPE="ATOR1TAPE"/>
</INSTANCE>

<INSTANCE ID="ATOR2IN" ACTOR="ATOR2" ORDER="2">
  <USE TAPE="ATOR2TAPE"/>
</INSTANCE>
```

Figura 6.43: Elemento *INSTANCE* para instanciar os atores e suas respectivas fitas de entrada

```

<applet
  WIDTH=360 HEIGHT = 320
  CODE="Animator.class"
  ARCHIVE="agal.jar, jdom.jar, jaxp.jar, crimson.jar"
  type="application/x-java-applet; version=1.3"
  scriptable="false"
  xml="ideia.xml"
  basearq="ideia/"
  useControls="yes">
</applet>

```

Figura 6.44: Exemplo de applet do sinal *Idéia*

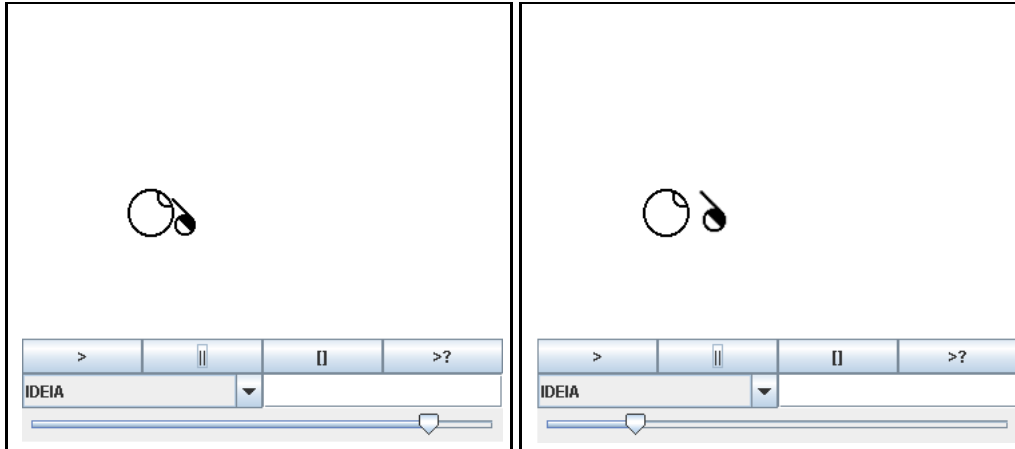


Figura 6.45: AGA Player da animação do sinal *Idéia*

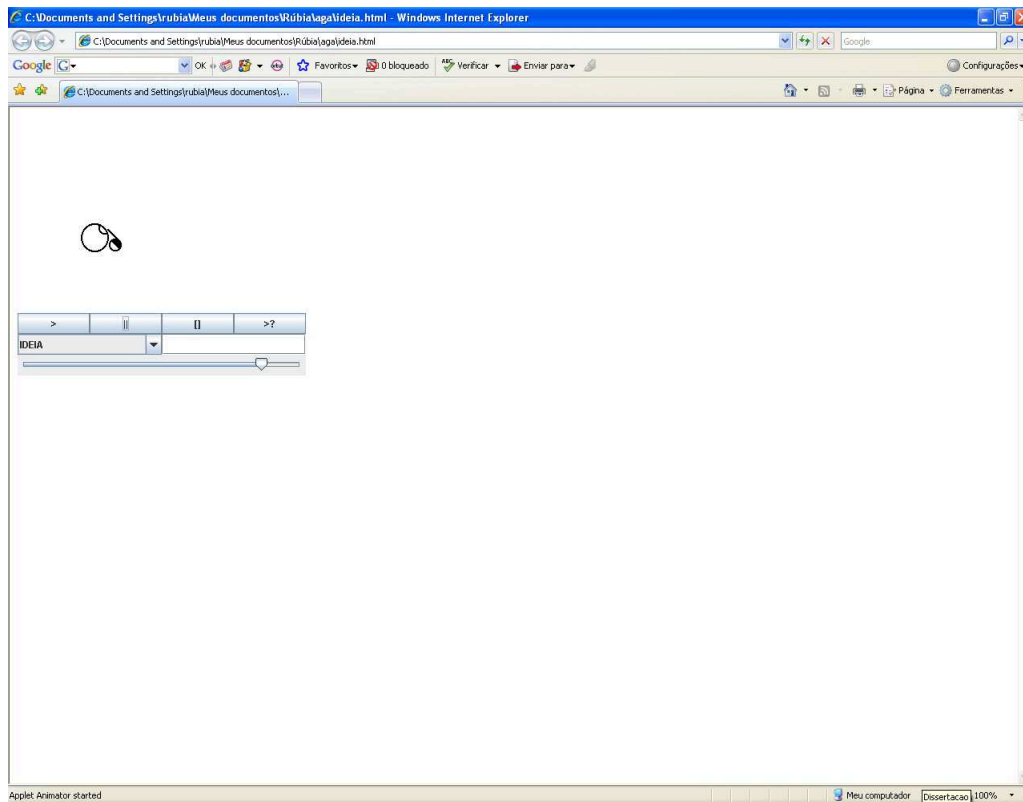


Figura 6.46: AGA Player da animação do sinal *Idéia*

Este documento AgaML, com as especificações dos atores, fitas de entrada e suas respectivas instâncias, serve de entrada para o AGA Player (Figura 6.45, 6.46), que é o programa de visualização do AGA na forma de applet JAVA (Figura 6.44).

## 7 CONCLUSÃO

Este trabalho está inserido no contexto de Processamento de Línguas Naturais (PLN) e envolve o estudo de Línguas de Sinais, sistema de escrita de sinais, ferramentas para suporte ao Processamento de Línguas de Sinais e por fim a animação de sinais através de textos. Ele está voltado à educação especial da comunidade surda, que utiliza línguas de modalidade gestual-visual em sua comunicação diária. A comunidade produz textos que são escritos através de notações gráficas que representam as partes do corpo envolvidas nos sinais, assim a animação é perfeitamente compreendida pelo surdo, já que sua língua natural é gestual-visual.

Os termos oral-auditivo e gestual-visual são diferenciados pela maneira como as línguas são produzidas e percebidas. Para as línguas orais utiliza-se o termo oral-auditivo e para as línguas de sinais, o gestual-visual onde gestual significa o conjunto de elementos lingüísticos manuais, corporais e faciais necessários para a articulação do sinal. Karnopp, em (CAMPOS; COSTA; STUMPF, 1996), cita que nas línguas de sinais, enquanto o emissor constrói uma sentença a partir desses elementos, o receptor utiliza os olhos ao invés dos ouvidos para entender o que está sendo comunicado. Desta forma, já que a informação lingüística é recebida pelos olhos, os sinais são construídos de acordo com as possibilidades perceptuais do sistema visual humano. Neste trabalho faz-se uma analogia entre a síntese da fala e animação de sinais (síntese de sinais).

Para este trabalho, a geração das animações foi feita a partir da Língua Brasileira de Sinais (LIBRAS) por se tratar da língua oficial dos surdos brasileiros. A LIBRAS possui estrutura gramatical própria e os sinais são formados por meio da combinação de formas e de movimentos das mãos e de pontos de referência no corpo ou no espaço. Segundo a legislação vigente, LIBRAS constitui um sistema lingüístico de transmissão de idéias e fatos, oriundos de comunidades de pessoas com deficiência auditiva do Brasil, na qual há uma forma de comunicação e expressão, de natureza visual-motora, com estrutura gramatical própria.

O AGA-Sign consiste em uma aplicação responsável pela animação de gestos <sup>1</sup> (ou sinais) voltados à Língua de Sinais e que pode ser usada tanto por pessoas surdas como por pessoas ouvintes. Esta aplicação possui construções especialmente projetadas para:

- Ensino/aprendizagem de deficientes auditivos;
- Ensino/aprendizagem de deficientes auditivos x ouvintes;
- Comunicação entre deficientes auditivos;
- Comunicação entre deficientes auditivos x ouvintes.

---

<sup>1</sup>Não se restringe à animação de sinais. O usuário pode produzir gestos que não constituem um sinal.

Para se chegar na animação de gestos (ou sinais) a partir de textos escritos em *SignWriting* usamos as seguintes ferramentas:

1. Editor de Textos *SignWriting* - a edição dos gestos (ou sinais) é feita a partir de símbolos visuais do sistema *SignWriting* de representação gráfica para línguas de sinais, como configurações das mãos, movimentos, expressões faciais e deslocamentos corporais.

O *SignWriting* é usado em mais de 25 países e no Brasil trabalha-se desde 1996, com a formação de vários grupos em escolas e associações de surdos interessados na difusão do sistema e na alfabetização de crianças surdas. Também, formaram-se grupos de pesquisa para auxiliar no desenvolvimento de ferramentas que façam uso do sistema. Em todos os lugares, o *SignWriting* tem despertado a atenção das chamadas comunidades surdas locais (estudantes, professores, intérpretes de língua de sinais, familiares de surdos, etc.) e, conforme alguns trabalhos publicados (STUMPF, 2000; CAPOVILLA et al., 1999; CAMPOS; GIRAFFA; SANTAROSA, 2000), o sistema tem tido uma boa aceitação por parte dessas comunidades surdas.

A aplicação do *SignWriting* à LIBRAS produzindo a escrita de sinais brasileira é uma ferramenta adequada para que os alunos surdos cumpram o objetivo de registrar por escrito sua língua visual.

2. Conversor *SW/SWML* - conversão de arquivos de sinais em arquivos texto. Um sinal convertido em texto *SWML* apresenta a categoria, grupo, número do símbolo, posição, rotação, variação, preenchimento e forma de cada símbolo, o Sinal-Símbolo-Sequência (SSS).

*SWML* é uma linguagem de marcação baseada em XML, que suporta adequadamente o armazenamento, troca e processamento de textos do *SignWriting*. Com a *SWML* é possível o intercâmbio de documentos entre diferentes programas, a análise de textos independente do editor e também serve como um formato de armazenamento de textos.

Quando os textos *SignWriting* forem produzidos usando-se o Editor *SignWriter* precisa-se converter os arquivos para *SWML*. Já os textos produzidos no editor *SWEdit* são salvos diretamente em *SWML*.

3. Tradutor de Textos *SWML/AgaML* - faz a leitura do SSS de cada símbolo do documento *SWML* para posterior tradução para um documento *AgaML*.

Para o desenvolvimento do Tradutor fez-se o estudo detalhado dos símbolos *SignWriting* para o conhecimento de suas características fonológicas e morfológicas (Capítulo 6), tornando possível a definição de algumas regras responsáveis pelo deslocamento dos símbolos durante o processo de animação. O Tradutor gera o documento *AgaML*, baseado nessas regras, contendo as especificações dos atores, fitas e instâncias *SignWriting*.

4. AGA - modelo de animação baseado na Teoria dos Autômatos.

O modelo AGA especifica a animação a partir de um conjunto de atores (objetos) e suas respectivas variações durante a animação. O modelo originou a linguagem de descrição da animação chamada *AgaML* (*AGA Markup Language*) (MAGALHÃES, 2002), que estrutura o conteúdo da animação em autômatos que descrevem



o comportamento de atores, onde cada ator é controlado por um autômato próprio. Os arquivos AgaML gerados pelo Tradutor são arquivos de entrada para o animador.

O estudo detalhado dos símbolos *SignWriting* foi fundamental para o processo de tradução de textos, possibilitando a identificação das características fonológicas e morfológicas dos sinais escritos e a determinação das regras de movimentos que controlam a posição e o deslocamento dos símbolos envolvidos na representação do sinal animado.

O Tradutor de Textos *SWML/AgaML* é a ferramenta que torna possível a animação dos sinais, pois faz a tradução de um sinal escrito em um sinal animado, através das regras de movimentos.

Em relação a adaptação do AGA à especificação de atores, fitas e instâncias dos símbolos *SignWriting*, o resultado foi satisfatório. Não houveram dificuldades neste sentido, pois provou-se ser possível animar os sinais através das regras de movimentos usadas pelo Tradutor de Textos *SWML/AgaML*.

Uma das limitações do AGA é a usabilidade de sua interface, onde o usuário tem que manipular diretórios e arquivos para fazer as animações. Com o desenvolvimento de uma interface para o AGA-Sign, o usuário não terá envolvimento com arquivos e diretórios. O usuário terá acesso a um editor de sinais para fazer a escrita e obterá a animação sem se envolver com a tradução (Tradutor *SWML/AgaML*). Assim, o AGA-Sign acaba com a limitação de usabilidade de interface do AGA.

As ferramentas que tornam possível a especificação do AGA-Sign, como a notação *SignWriting*, os editores *SignWriting* e a linguagem *SWML* foram e são usadas em algumas pesquisas voltadas a área de desenvolvimento de softwares que contribuem para o aprendizado da língua de sinais (INSTITUTE, 2004; COSTA, 2006f; CAMPOS; GIRAFFA; SANTAROSA, 2000). Assim, espera-se o mesmo do AGA-Sign, e que possa contribuir com o aprendizado da língua de sinais no que diz respeito à escrita pelo fato da aplicação ser um conjunto de ferramentas nas quais o aluno pode produzir os sinais, armazená-los em dicionários (editor) e animá-los (AGA-Sign), auxiliando na prática da escrita de sinais e a familiarização com a língua.

O aprendiz da língua de sinais também poderá usar o animador como um verificador de correção de sinais: na dúvida se o sinal escrito é ou não o que está querendo representar, o usuário animará o sinal para ter a confirmação.

O AGA-Sign poderá também ser incorporado a editores de textos *SignWriting* e dicionários, como por exemplo ao *SWEdit* e ao *SignPuddle* (disponível em ambiente Web (DAC, 2006)). Assim o usuário fará a escrita e obterá a animação de seus sinais no mesmo ambiente de desenvolvimento.

Para este trabalho foram estudados e criadas regras para símbolos de movimentos retilíneos e circulares, que cobrem cerca de 30% dos símbolos mais representativos em sinais na LIBRAS.

## 7.1 Trabalhos Futuros

No trabalho, na fase atual, a animação é feita por sinal e não por sentença (frase). A animação de sinais em sequência é um trabalho futuro.

Como trabalhos futuros está programado o estudo dos símbolos de movimentos curvos, movimentos de ombros e movimentos de cabeça para a geração de suas regras de animação.

Sugere-se o desenvolvimento de uma interface de usuário para o AGA-Sign, para que o usuário não tenha envolvimento com arquivos e diretórios da interface do AGA.

Recentemente foi desenvolvido pelo grupo o AGA-J (MARTINS et al., 2004) para animações em 3D e como trabalho futuro poderiam ser feitas as adaptações para as animações de sinais 2D para 3D.

## REFERÊNCIAS

- ACCORSI, F. **Animação Bidimensional para World Wide Web Baseada em Autômatos Finitos**. 2002. Dissertação (Mestrado em Ciência da Computação) — PPGC/UFRGS, Porto Alegre.
- AJA. **Língua de Sinais - LIBRAS**. Disponível em: <<http://www.libras.org.br/indexPort.html>>. Acesso em: jan. 2004.
- BAKER, C.; BATTISON, R. (Ed.). **Signs Have Parts: a simple idea**. [S.l.]: MD: National Association of the Deaf, 1980. p.35–51.
- BENTELE, S. **About The HamNoSys System**. Disponível em: <<http://www.signwriting.org/forums/linguistics/ling007.html>>. Acesso em: mar. 2005.
- BESSA, M. **Educação em LIBRAS: há 145 anos, o ines promove educação para surdos**. Disponível em: <<http://www.educacaopublica.rj.gov.br/jornal/materia.asp?seq=200>>. Acesso em: mai. 2004.
- CAMPOS, M. d. B.; COSTA, A. C. d. R.; STUMPF, M. Sistema de representação interna e externa das línguas de sinais. In: CONGRESO IBEROAMERICANO DE COMUNICACIÓN ALTERNATIVA Y AUMENTATIVA, 3., 1996, Viña del Mar, Chile. **Anais...** [S.l.: s.n.], 1996.
- CAMPOS, M. d. B.; GIRAFFA, L. M. M.; SANTAROSA, L. M. C. SIGNSIM: uma ferramenta para auxílio à aprendizagem da língua brasileira de sinais. In: CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, RIBIE. 5., 2000, Viña del Mar - Chile. **Anais...** [S.l.: s.n.], 2000.
- CAPOVILLA, F. C.; CAPOVILLA, A. G. S. Leitura de estudantes surdos: desenvolvimento e peculiaridades em relação à de ouvintes. In: **Educação Temática Digital**. Campinas, SP: [s.n.], 2006. v.7, n.2, p.217 – 227.
- CAPOVILLA, F. C.; CAPOVILLA, A. G. S.; VIGGIANO, O. K.; BIDÁ, M. C. P. R. Avaliando compreensão de sinais da Libras em escolares surdos do ensino fundamental. In: **Interação**. Curitiba, PR: [s.n.], 2004. v.8, n.2, p.159 – 169.
- CAPOVILLA, F. C.; DUDUCHI, M.; RAPHAEL, W. D.; LUZ, R. D.; ROZADOS, D.; CAPOVILLA, A. G. S.; MACEDO, E. C. Brazilian Sign Language lexicography and technology: dictionary, digital encyclopedia, chereme-based sign-retrieval and quadriplegic deaf communication systems. In: **Sign Language Studies**. Washington, DC, USA: [s.n.], 2003. v.3, n.4, p.393 – 430.

CAPOVILLA, F. C.; MAZZA, C. Z.; AMENI, R.; NEVES, M. V.; CAPOVILLA, A. G. S. Quando surdos nomeiam figuras: processos quirêmicos, semânticos e ortográficos. In: **Perspectiva**. Florianópolis, SC: [s.n.], 2006. v.24, p.153 – 175.

CAPOVILLA, F. C.; RAPHAEL, W. D.; SHIN, S. Y.; ROCHA, M. P.; GERALDES, A.; MARQUES, S.; LUZ, R. D.; NEVES, S. L. G.; VIGGIANO, K. Dicionário de língua brasileira de sinais: ilustração e escrita visual direta de 3500 sinais usados pelos surdos em são paulo. In: JOURNAL OF THE NATIONAL INSTITUTE OF EDUCATION OF THE DEAF, 1999, Brazilian Government. **Anais...** [S.l.: s.n.], 1999. v.11, p.84–88.

COSTA, A. C. d. R. **SWML - SignWriting Markup Language**. Disponível em: <<http://sign-net.ucpel.tche.br/swml/>>. Acesso em: jun. 2006.

COSTA, A. C. d. R. **Aplicações de Sign Writing**. Disponível em: <<http://rocha.ucpel.tche.br/SignWriting/SWProj.html>>. Acesso em: jun. 2006.

COSTA, A. C. d. R. **Dicionário Português-LIBRAS-English**. Disponível em:<<http://rocha.ucpel.tche.br/signwriting/dicionario/Dic-Port-LIBRAS-Engl.htm>>. Acesso em: jun. 2006.

COSTA, A. C. R. **Lições sobre o SignWriting**. Disponível em: <<http://sign-net.ucpel.tche.br/licoes-sw/licoes-sw.pdf>>. Acesso em: jun. 2006.

COSTA, A. C. R. **Site do Projeto SignNet**. Disponível em: <<http://sign-net.ucpel.tche.br/resumo-executivo.htm>>. Acesso em: jan. 2006.

COSTA, A. C. R. **Um convite ao Procesamento de Línguas de Sinais**. Palestra Convidada no 2º Workshop sobre Tecnologia da Informação e da Linguagem Humana. Disponível em: <<http://rocha.ucpel.tche.br/TIL2004/til-2004.pdf>>. Acesso em: jun. 2006.

DAC, D. A. C. f. S. **SignSpelling Guidelines 2004**. Disponível em: <<http://www.signwriting.org/archive/list/list100.html>>. Acesso em: jan. 2004.

DAC, D. A. C. f. S. **SignWriter DOS v.4.4**. Disponível em: <<http://www.signwriting.org/forums/software/sw44/index.html>>. Acesso em: jan. 2004.

DAC, D. A. C. f. S. **SignWriter 5.0 in Java**. Disponível em: <<http://www.signwriting.org/forums/software/sw50/index.html>>. Acesso em: jan. 2004.

DAC, D. A. C. f. S. **SignPuddle**. Disponível em:<<http://signbank.org/signpuddle/sgn-BR/index.php>>. Acesso em: jun. 2006.

DUTOIT, T. **An Introduction to Text-to-Speech Synthesis**. Dordrecht: Kluwer Academic, 1997. 312p.

FENEIS. **LIBRAS**. Disponível em: <<http://www.feneis.com.br>>. Acesso em: maio. 2005.

FIAP. **Projeto de olho na WEB!** [S.l.]: Faculdade de Informática e Administração Paulista. Disponível em: <[http://www.fiap.com.br/portal/int\\_cda\\_conteudo.jsp?ID=9647](http://www.fiap.com.br/portal/int_cda_conteudo.jsp?ID=9647)>. Acesso em: out. 2004.

GERMAN SIGN LANGUAGE, I. of; DEAF, C. of the. **HamNoSys 3.0 - Example**. Disponível em: <<http://www.sign-lang.uni-hamburg.de/Projekte/HamNoSys/HamNoSysErklaerungen/englisch/02ExampleTiring.html>>. Acesso em: mar. 2005.

HANKE, T. **HamNoSys - The Hamburg Notation System**. Disponível em: <<http://sign-lang.uni-hamburg.de/Projects/HamNoSys.html>>. Acesso em: mar. 2005.

HOPCROFT, J. E.; ULLMAN, J. D. **Introduction to Automata Theory, Languages and Computation**. [S.l.]: Addison-Wesley, 2000. 42-45p.

INSTITUTE, I. . T. **VSigns**. Disponível em: <http://vsigns.iti.gr>. Acessado em abr, 2004.

MAGALHÃES, G. d. C. Aga Player: animação 2d baseada em autômatos para a web. In: 2002, Porto Alegre. **Anais...** [S.l.: s.n.], 2002. Instituto de Informática, Universidade Federal do Rio Grande do Sul. Trabalho de Conclusão.

MARTIN, J. **About The Stokoe System**. Disponível em: <http://www.signwriting.org/forums/linguistics/ling006.html>. Acessado em março, 2005.

MARTINS, J. D. F.; NEDEL, L. P.; MENEZES, P. F. B.; ACCORSI, F. An Automata based Animation Engine to Control Articulated Characters. In: SVR IEEE COMPUTER SOCIETY, 2004, São Paulo. **Anais...** [S.l.: s.n.], 2004. p.315-326.

MARTINS, R. T.; NUNES, M.; C., D. D. S. B.; RINO, L. H. M.; OLIVEIRA JR., O. N.; MONTILHA, G. Introdução ao processamento das línguas naturais. In: NOTAS DIDÁTICAS ICMC, 1999, 1999, São Carlos, São Paulo. **Anais...** [S.l.: s.n.], 1999.

MENEZES, P. B. **Hyper Seed**. Disponível em: <<http://teia.inf.ufrgs.br/hyper-automaton.html>>. Acesso em: jan. 2004.

MENEZES, P. B. **Linguagens Formais e Autômatos**. Porto Alegre: Sagra Luzatto, 2005.

STOKOE, W.; CASTELINE, D. C.; CRONEBERG, C. G. **A Dictionary of American Sign Language on Linguistic Principles**. Washington: Gallaudet College Press, 1965.

STUMPF, M. R. Língua de Sinais: escrita dos surdos na internet. In: RIBIE - V CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 2000, 2000, Viña del Mar - Chile. **Anais...** [S.l.: s.n.], 2000.

SUTTON, V. **Sign Writing**. Disponível em: <<http://www.signwriting.org>>. Acesso em: jun. 2005.

SUTTON, V. **SignBank Databases**. Disponível em: <<http://www.signbank.org/signbank.html>>. Acesso em: abr. 2006.

TORCHELSEN, R. P.; COSTA, A. C. R.; DIMURO, G. P. Editor para textos em línguas de sinais escritos em Sign Writing. In: XXVIII CONFERENCIA LATINOAMERICANA DE INFORMÁTICA, 2002, Montevideu - Uruguai. **Anais...** [S.l.: s.n.], 2002. v.1, p.1-10.

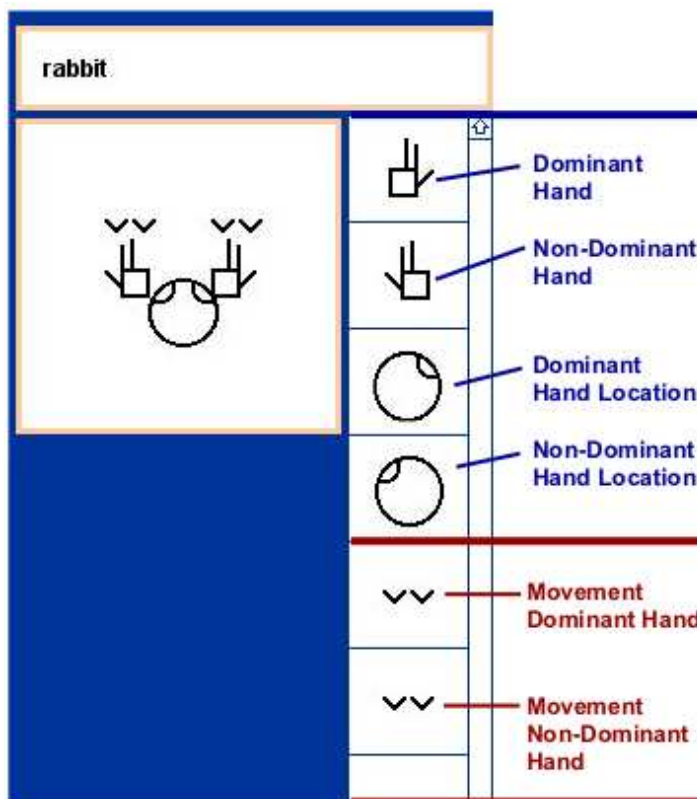
## ANEXO A EXEMPLOS DE SOLETRAÇÕES

### A.1 Quando não existe mão dominante

Mesmo que as mãos sejam iguais, e uma não esteja dominando a outra, tem-se que escolher uma ou outra para ser a primeira. Mantendo, assim, a escolha consistente com as outras entradas no dicionário.

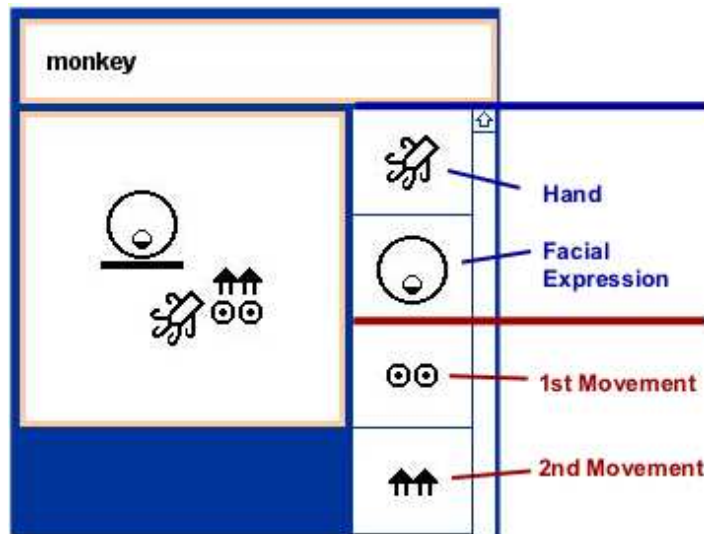
### A.2 Movimentos dominantes e não-dominantes

O movimento para a mão dominante é colocado primeiro, e o movimento para a mão não-dominante é colocado em segundo.



### A.3 Expressões Faciais

Quando uma expressão facial dá o verdadeiro significado a um sinal ela deve ser incluída na soletração.



### A.4 Guia de Pronúncia

*SignWriting* pode ser classificado por detalhados marcadores de posição que não são escritos, atualmente nos próprios sinais. Os marcadores mostram a localização precisa do contato no corpo ou no espaço. E também podem mostrar a um leitor do dicionário como produzir um sinal corretamente.

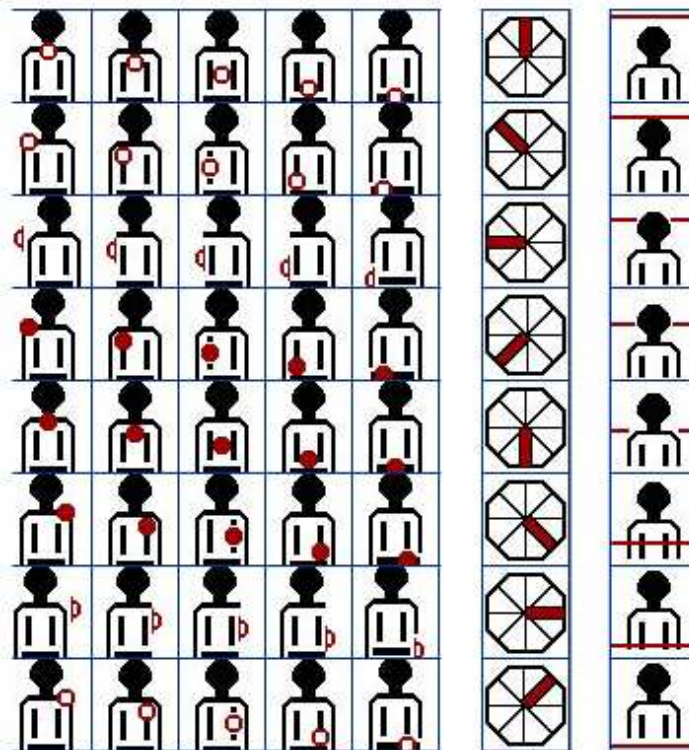


Figura A.1: Alguns marcadores de posição disponíveis para pesquisa (DAC, 2004a)



## ANEXO B DOCUMENTO SWML

Documento *SWML* correspondente ao sinal “mãe”.



Figura B.1: Sinal “mãe”

```
<?xml version="1.0"?>
<swml version="1.0"file-type="text"xmlns="http://www.ucpel.tche.br/2002/05/swml"
xmlns:xlink="http://www.w3.org/1999/xlink"xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"xsi:schemaLocation="http://www.ucpel.tche.br/2002/05/
swmlhttp://swml.ucpel.tche.br/schemas/swml/2002/05/swml.xsd»
  <generator>
    <name>SW-Edit</name>
    <version>1.1</version>
  </generator>
  <text symbol-set="sss-2002»
    <defaults>
      <unit>pt</unit>
      <page-height>1200</page-height>
      <page-width>750</page-width>
      <line-height>100</line-height>
      <cell-bgcolor>255,255,255</cell-bgcolor>
      <corners-color>0,0,0</corners-color>
      <txt-color>0,0,0</txt-color>
      <txt-font>
        <name>Times New Roman</name>
        <size>12</size>
        <family>wxROMAN</family>
        <style>wxNORMAL</style>
        <weight>wxNORMAL</weight>
        <underlined>0</underlined>
      </txt-font>
    </defaults>
  <page height="1200"width="750»
```

```

<line x="0"y="0"height="150"width="750»
  <cell x="0"y="0"height="150"width="150"bgcolor="255,255,255"cornersColor="0,0,0»
    <signbox>
      <symb x="35"y="47"x-flop="0"y-flop="0"color="0,0,0»
        <category>03</category>
        <group>05</group>
        <symbnum>005</symbnum>
        <variation>01</variation>
        <fill>01</fill>
        <rotation>01</rotation>
      </symb>

      <symb x="68"y="45"x-flop="0"y-flop="0"color="0,0,0»
        <category>02</category>
        <group>01</group>
        <symbnum>001</symbnum>
        <variation>01</variation>
        <fill>01</fill>
        <rotation>01</rotation>
      </symb>

      <symb x="88"y="44"x-flop="0"y-flop="0"color="0,0,0»
        <category>02</category>
        <group>01</group>
        <symbnum>001</symbnum>
        <variation>01</variation>
        <fill>01</fill>
        <rotation>01</rotation>
      </symb>

      <symb x="51"y="65"x-flop="0"y-flop="0"color="0,0,0»
        <category>01</category>
        <group>01</group>
        <symbnum>001</symbnum>
        <variation>01</variation>
        <fill>02</fill>
        <rotation>02</rotation>
      </symb>
    </signbox>
  </cell>
</line>
</page>
</text>
</swml>

```