

Ferramentas Auxiliares para Criação de Arquivos Descritivos de Sinais de Libras

Palavras-Chave Acessibilidade - Libras - Computação Gráfica

Departamento de Engenharia da Computação e Automação Industrial - FEEC - UNICAMP

Autor: Leandro Martin Guertzenstein Angare (leangare@hotmail.com) | Orientador: Prof. Dr. José Mario De Martino (martino@dca.fee.unicamp.br)

<Introdução>

A acessibilidade em ambientes computacionais é um problema que tem recebido grande atenção ultimamente, em especial com o aumento da utilização de plataformas virtuais no ensino. O projeto do Agente Virtual Sinalizador de Libras se apresenta como uma destas propostas ao apresentar uma forma de comunicação alternativa para apresentação de conteúdo à usuários com deficiência auditiva.

O trabalho de Iniciação Científica aqui apresentado insere-se no grupo de pesquisa que propõe a criação de um Agente Virtual (Figura 1) capaz de reproduzir frases em Libras; para tanto foram pesquisadas por Amaral as complexas configurações de mão e movimentações que compõe cada sinal, formando bancos de arquivos capazes de serem combinados afim de formar, a partir de descrições mínimas de configuração de mão, uma sentença. Para tanto, este trabalho se propôs a desenvolver uma solução para a criação de dois tipos de arquivos: arquivos XML descritivos de configuração de mão e tabelas CSV para posicionamento destas no espaço (Figura 2).

Para a criação de tais arquivos haviam duas alternativas: colocar os valores nos arquivos e testar o posicionamento; ou alterar dentro do software proprietário (Maya) as juntas necessárias e copiar os valores, manualmente. Ambas soluções dispendem demasiado tempo de execução e inibiriam a criação de novos arquivos por outros usuários interessados em utilizar e expandir o sistema. Essa foi a motivação para o desenvolvimento das ferramentas que compõe este projeto de Iniciação Científica.



FIGURA 1

Para a implementação da ferramenta foi definida a utilização da linguagem Python em detrimento de C++ e MEL (Maya Embedded Language) pela sua capacidade de trabalhar com programação orientada a objetos (impossível em MEL) e portabilidade.

Foi utilizada a metodologia clássica de desenvolvimento de sistemas com o paradigma de prototipação pois existia a possibilidade de contato constante e próximo com o usuário, a aluna Wanessa Amaral, que demandava o resultado das ferramentas, documentos XML e tabela CSV, para a implementação do Agente. Com a abordagem de prototipação foi possível construir as ferramentas ao mesmo tempo em que elas eram testadas/validadas na criação dos documentos, possibilitando também a utilização destes documentos no projeto de doutorado da referida aluna.

<Resultados>

Como resultado do projeto, foram desenvolvidas duas ferramentas em Python que permitem a criação de documentos XML e tabelas CSV a partir de informações de dentro do software proprietário Autodesk Maya (Figura 3).

Tais ferramentas foram utilizadas para gerar um banco de arquivos de descrição das 26 letras do alfabeto latino (XML) e um arquivo txt (tabela CSV) contendo informações de posicionamento de mão no espaço (Figura 4).

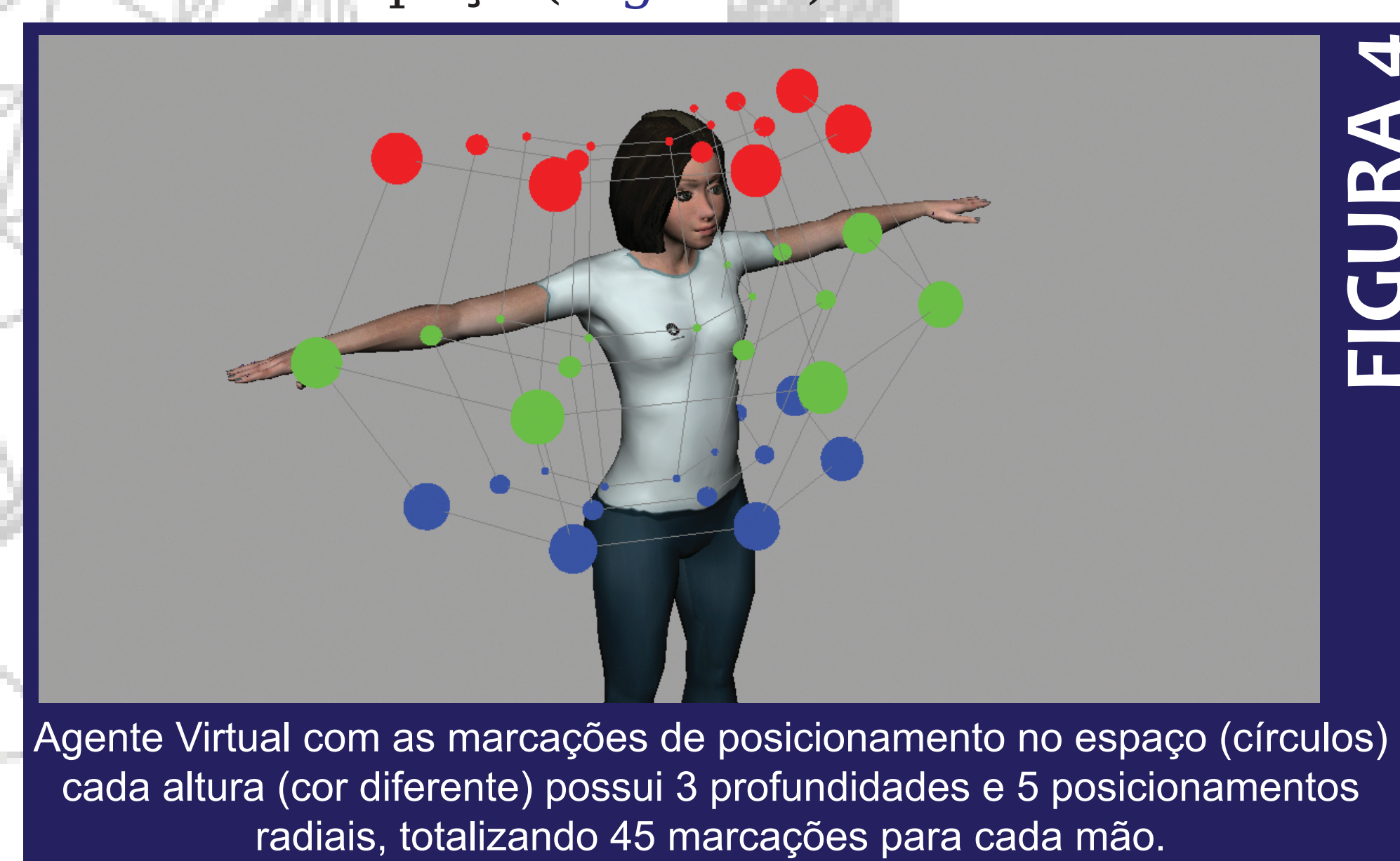


FIGURA 4

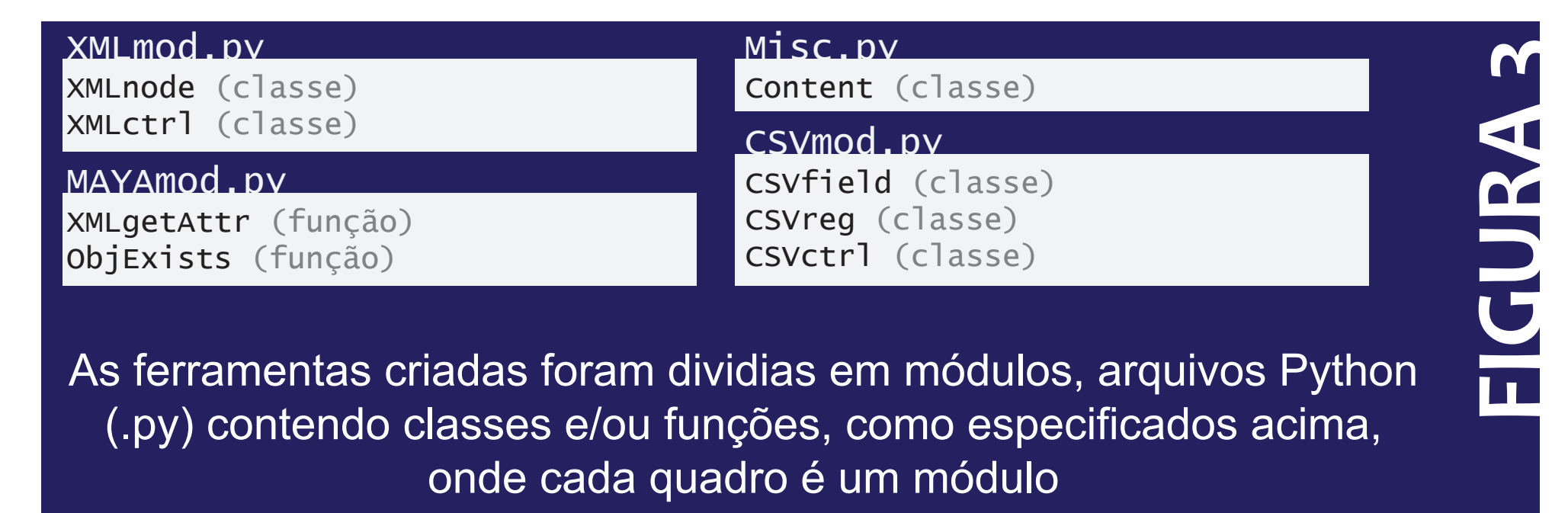


FIGURA 3

<Conclusão>

O projeto obteve êxito ao criar as ferramentas a que se dispôs. Uma das características almeçadas e alcançada foi a portabilidade entre diferentes softwares, que é possível através do encapsulamento de funções específicas ao software de animação e modelagem 3D no módulo referente a ele como mostrado na Figura 3 (no caso, o software Maya possui seu módulo -MAYAmo.py)

Outro ponto importante foi a criação de um validador Schema [7] para os arquivos XML, principalmente para impedir rotações indevidas de juntas, como definido em [8].

<Bibliografia>

- [1] Amaral, W. M.; De Martino, J. M. **Modelo de transcrição da Língua de Sinais Brasileira voltado a implementação de agentes virtuais sinalizadores.**
- [2] Amaral, W. M.; Angare, L.; Bezerra, J; De Martino J. M.; Franchi Jr., G. O. **Agente 3D Sinalizador de Libras.**
- [3] GOULD, David A. D. **Complete Maya Programming: An Extensive Guide to MEL and C++ API.**
- [4] LUTZ, Mark. **Learning Python.**
- [5] RAY, Erik T. **Learning XML: Creating Self-Describing Data.**
- [6] W3C. **Extensible Markup Language (XML)**, disponível em <http://www.w3.org/XML/>.
- [7] W3C. **XML Schema**, disponível em <http://www.w3.org/XML/Schema>.
- [8] KAPANDJI, A. I. **Fisiologia Articular: Esquemas Comentados de Mecânica Humana**

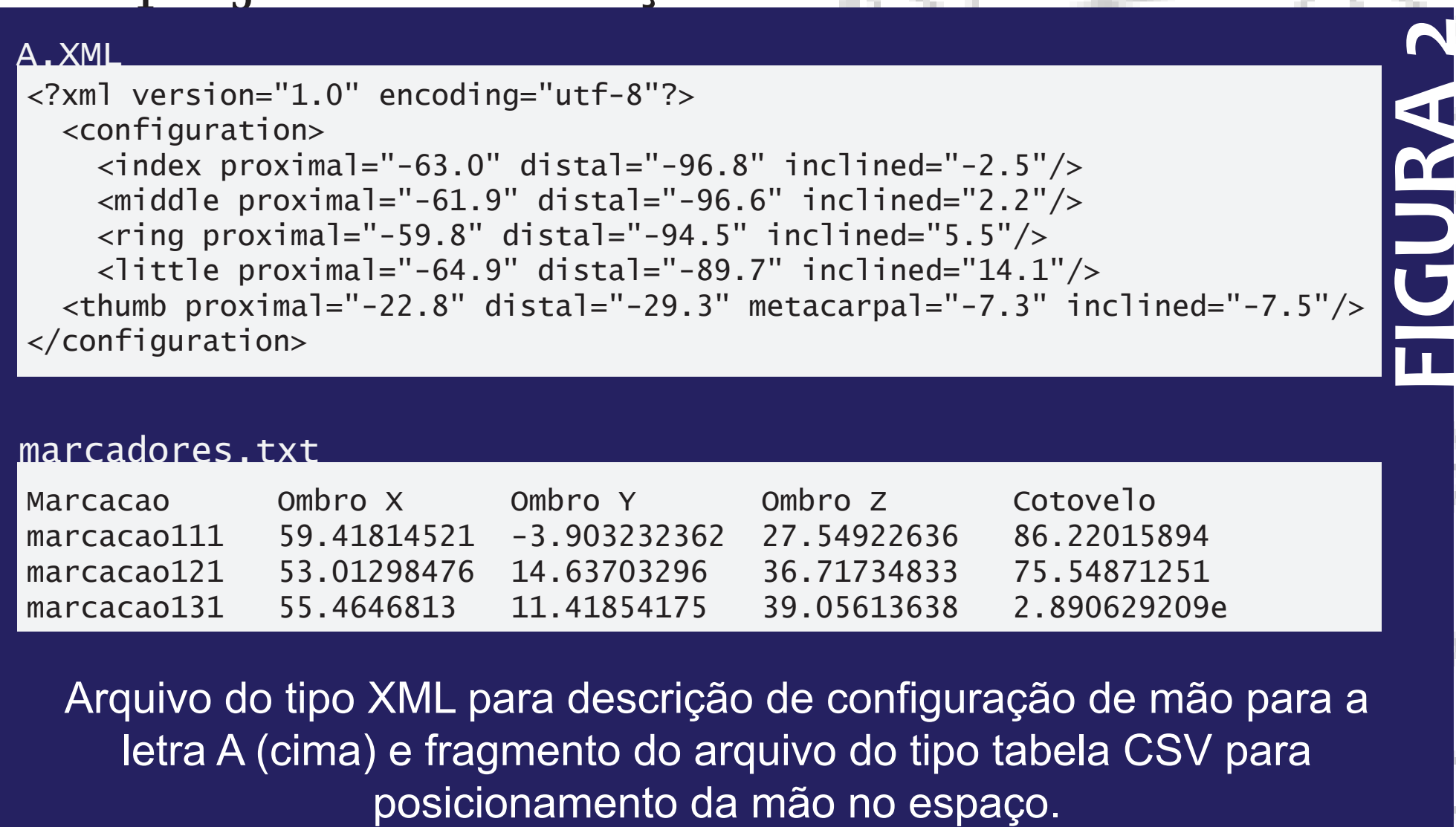


FIGURA 2

<Metodologia>

Inicialmente foi necessário o levantamento das estruturas dos arquivos (XML e CSV) utilizando como principal referências [5] e [6]. No caso de tabelas do tipo CSV existem basicamente duas estruturas: os registros e os campos que os compõem.

Para os arquivos XML, temos diversas estruturas, sendo a principal a chamada entidade. Tais estruturas nortearam a criação das classes utilizadas nas ferramentas, juntamente com o documento de requisitos.