

**TSL**

**Uma linguagem para especificação de tradutores**

Marcello Novaes de Amorim

Dissertação de Mestrado em Informática

Mestrado em Informática  
Universidade Federal do Espírito Santo

Vitória – ES. 2007

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Marcello Novaes de Amorim

**TSL**

**Uma linguagem para especificação de tradutores**

Dissertação apresentada como requisito parcial à  
obtenção do grau de Mestre em Informática.  
Programa de Pós-Graduação em Informática.

Universidade Federal do Espírito Santo.

Orientador: Prof. Dr. Orivaldo de Lira Tavares.

Vitória – ES. 2007

Dados Internacionais de Catalogação-na-publicação (CIP)  
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

---

Amorim, Marcello Novaes de, 1979-

A524t TSL, uma linguagem para especificação de tradutores / Marcello Novaes de Amorim. – 2007.

109 f. : il.

Orientador: Orivaldo de Lira Tavares.

Dissertação (mestrado) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Serviços de tradução. 2. Tradução mecânica. 3. Tradutores (Programas de computador). 4. Ontologia. 5. Sistemas de memória de computadores. 6. Serviços na Web. I. Tavares, Orivaldo de Lira. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 004

---

# **TSL**

Uma linguagem para especificação de tradutores

## **Marcello Novaes de Amorim**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Aprovada em 26 de julho de 2007.

### **COMISSÃO EXAMINADORA**

---

**Prof. Dr. Orivaldo de Lira Tavares**  
**Universidade Federal do Espírito Santo**  
**Orientador**

---

**Prof. Dr. José Gonçalves Pereira Filho**  
**Universidade Federal do Espírito Santo**  
**Examinador Interno**

---

**Prof. Dr. Luis Cláudius Coradine**  
**Universidade Federal de Alagoas**  
**Examinador externo**

---

**Profa. Dra. Aucione Smarsaro**  
**Universidade Federal do Espírito Santo-DLL**  
**Examinadora externa**

## **Agradecimentos**

Agradeço à Dani, minha noiva, por toda a paciência comigo durante esse tempo e por todo o apoio na minha opção pela vida acadêmica. Por ter me animado e motivado a concluir minha graduação, e principalmente, por me ajudar a descobrir quem sou e a me tornar um homem melhor.

À minha mãe, pelo suporte que me possibilitou maior dedicação aos estudos. Por todo cuidado e carinho, por aceitar e apoiar minhas decisões.

Ao meu orientador Orivaldo, pela confiança e credibilidade depositadas em mim. Por me inspirar a escolher a carreira docente, pela ajuda, compreensão e paciência. Tornou-se um grande amigo e um modelo para mim, como educador e como ser humano.

Aos meus colegas de mestrado, pelo apoio e ajuda, Viviane, Wesley, Vitor, Bernard e principalmente ao Eduardo, pela colaboração durante o processo de escrita da dissertação e ao professor José Gonçalves por aceitar o convite de última hora para participar da banca avaliadora.

*“O tempo é apenas uma ilusão criada para justificar as discrepâncias entre nosso estado atual de percepção e nossas lembranças”*

Douglas Adams [Guia 1979]

## Glossário de Termos

<b>Termo</b>	<b>Definição</b>
Ação Semântica	Conjunto de operações aplicadas sobre um padrão gramatical durante processos de tradução.
Análise Lexica	Processo de analisar linhas de caracteres e produzir uma seqüência de símbolos léxicos que podem ser manipulados mais facilmente por um Parser.
Análise Sintática	Processo de analisar uma seqüência de entrada para determinar sua estrutura gramatical segundo uma determinada gramática formal.
Aplicação	Conjunto de programas para computador.
Backtracking	Procedimento aplicado em buscas em profundidade que permite que se retorne a um nó anterior, descendo por um ramo diferente da árvore, caso se deseje encontrar um resultado diferente.
Base de Conhecimento	Equivalente à Base de Dados, porém, tem como finalidade armazenar conhecimento de maneira utilizável pelos Sistemas Baseados em Conhecimento.
Base de Dados	Conjuntos de dados com uma estrutura regular que organizam informação.
Caso	Um caso TSL é uma generalização que pode ser tanto um Fato quanto uma Regra.
Código Aberto	Programa de computador cujo código fonte é disponível e visível publicamente.
Código Fonte	Conjunto instruções escritas de maneira lógica em uma linguagem de programação.
Conhecimento Explícito	Conhecimento que é registrado de alguma maneira que possa ser acessado.
Conhecimento Tácito	Conhecimento adquirido por uma pessoa durante suas experiências individuais.
Documento	Agrupamento de registros que seguem uma regra estrutural, e que contém informações.
Documento Multimídia	Documento formado composto por objetos multimídia.
Domínio	Refere-se a um domínio do conhecimento. Pode ser algo mais geral como um idioma, ou um sub-conjunto do idioma, assim como pode se referir a uma área específica. Ex: Português, Inglês Médico, Física Quântica.
E-Mail	Correio eletrônico, um endereço utilizado para enviar e receber mensagens pela WEB.
Especificação Declarativa	Conjunto de regras que definem um programa computacional de acordo com a Programação Declarativa.
Estrutura de Dados	Forma de organizar e agrupar coerentemente os dados.
Fato	Registro TSL capaz de reconhecer padrões fixos (totalmente instanciados) em um texto, ou seja, reconhece padrões formados apenas por palavras.



GNU	Projeto que tem como objetivo criar sistemas computacionais totalmente livre, que podem ser utilizado e distribuído sem custos sobre licenças.
HTML	HyperText Markup Language. Principal linguagem de marcação usada na criação das páginas de Internet.
Implementação	Fase do Ciclo de Vida de um programa computacional que corresponde à elaboração e preparação dos módulos necessários à sua execução.
Interface WEB	Interface com o usuário, baseada em páginas de Internet.
ISO	International Standardization Organization. Organização internacional que aprova normas internacionais em todos os campos técnicos, exceto na eletricidade e eletrônica.
Linguagem de Marcação	Conjunto de códigos aplicados a um texto ou a dados, com o fim de adicionar informações particulares sobre eles.
Linguagem de Programação	Conjunto padronizado de regras sintáticas e semânticas usadas para expressar as instruções que definem um programa de computador.
Links	Ligação entre documentos que podem ser, entre outras coisas, páginas de Internet.
LISA	The Localization Industry Standards Association. Órgão dedicado a criar padrões para compatibilidade.
Login	Apelido utilizado para identificar um usuário em um ambiente computacional.
Memória de Tradução	Método de tradução que visa preservar a integridade semântica.
Modelo Mental	Cena formada mentalmente por uma pessoa, baseado em seu conhecimento e estado mental, ao receber uma mensagem.
NULL	Valor nulo. Indica que um determinado campo de armazenamento tem contem nenhum valor.
Objeto	Entidade que pode ser física, conceitual em um programa.
Objeto Multimídia	Documentos de diferentes naturezas, como textos, vídeos, imagens e sons.
On-Line	Adjetivo que define positivamente o acesso de algo por meio de uma rede.
Orientação a Objetos	Paradigma de programação baseado na composição e interação entre diversas unidades chamadas de objetos.
OSCAR	Open Standards for Container/Content Allowing Re-use. Setor do LISA responsável pela criação do TMX.
Parser	Conjunto de procedimentos que tem como objetivo analisar um conjunto de símbolos léxicos para extrair as informações contidas nele.
Programa	Listas de instruções para o computador executar.
Programação Declarativa	Paradigma de programação baseado na descrição dos procedimentos e não como eles funcionam.
Programação Procedural	Paradigma baseado em chamadas de procedimentos.
Regra	Registro TSL capaz de reconhecer em um texto padrões formados por palavras e classes gramaticais.
Serviço	Programa ao qual um Átomo de Tradução delega o processamento de padrões.

SGML	Standard Generalized Markup Language. Metalinguagem através da qual se pode definir Linguagens de Marcação para documentos.
Sistema Baseado em Conhecimento	Sistemas computacionais que utilizam bases de conhecimento para realizar inferências.
Sistema de Produção	Programa que processa um texto de acordo com pares do tipo condição-ação.
Sistema de Produção	Sistema que processa um texto de acordo com pares do tipo condição-ação.
Software Livre	Programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído com algumas restrições.
TA	Átomo de Tradução. Unidade contida em Casos TSL e representa.
Tag	São estruturas de linguagem de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim.
TDS Tradução dirigida por sintaxe	Método de tradução baseado na identificação de regras gramaticais.
TMX	Translation Memory Exange. Formato baseado na XML utilizada para armazenar e compartilhar Memórias de Tradução.
TSL	Translator Specification Language. Linguagem Declarativa baseada no TMX, utilizada para especificar tradutores baseados em conhecimento.
TU	Unidade de Tradução. Registro do TMX que representa uma entrada na Memória de Tradução.
TUV	Variante da Unidade de Tradução. Parte de um TU que indica sua representação em um determinado idioma.
Usuário	Pessoa que utiliza um programa computacional.
Vetor	Estrutura de dados capaz de armazenar múltiplos valores e conta com um índice que permite o acesso direto a cada uma de suas posições.
W3C	O World Wide Web Consortium é um consórcio de empresas que desenvolve tecnologias denominadas padrões da web.
WEB	Abreviação de WWW (World Wide Web), rede global de comunicação que forma a Internet.
WebServices	Componentes que permitem a integração de aplicações por meio de troca de dados pela WEB.
XML	eXtensible Markup Language. Recomendação da W3C para gerar linguagens de marcação para necessidades especiais.

## **Resumo**

*A TSL (Translator Specification Language) é uma linguagem declarativa, que permite a um especialista representar o conhecimento necessário para traduzir mensagens associadas ao seu domínio de conhecimento. Dessa forma, podem ser geradas traduções semelhantes às que seriam criadas manualmente por ele.*

*Ao se identificar os padrões formados pelo elemento de uma mensagem e aplicar sobre cada padrão ações descritas na especificação TSL, é possível gerar uma tradução, ou seja, uma nova representação dos conceitos e relacionamentos contidos na mensagem original.*

*Uma vez que a TSL torna possível o processamento individual de padrões identificados em uma mensagem, é possível especificar programas baseados na busca e aplicação de ações semânticas sobre eles.*

## **Abstract**

*The TSL (Translator Specification Language) it is a declarative language, which allows a specialist to represent the necessary knowledge to translate messages associated to one's knowledge domain. This way, it's possible to generated translations similar to the ones created by one.*

*While identifying the generated patterns by the elements of a message and applying on each pattern actions described in the TSL's specification, it's possible to generate a translation, which means, a new representation of the concepts and relations contained in the original message.*

*Since the TSL allows processing individual fragments of a message, it's possible to specify programs that searching patterns and applying semantic actions to it.*

## Lista de Figuras

<i>Figura 2.1 - Relacionamento entre os diversos métodos de tradução.....</i>	<i>19</i>
<i>Figura 2.2 – Interface WEB do Babel Fish.....</i>	<i>27</i>
<i>Figura 2.3 - Selecionando os idiomas de e para no Babel Fish.....</i>	<i>28</i>
<i>Figura 2.4 – Texto exemplo traduzido no Babel Fish.....</i>	<i>29</i>
<i>Figura 2.5 – Selecionando Origem/Destino no Power Translator.....</i>	<i>31</i>
<i>Figura 2.6 – Entrando com um texto no Power Translator.....</i>	<i>32</i>
<i>Figura 2.7 – Selecionando uma frase no Power Translator.....</i>	<i>33</i>
<i>Figura 2.8 – Traduzindo uma frase no Power Translator.....</i>	<i>34</i>
<i>Figura 2.9 – Traduzindo um texto no Power Translator.....</i>	<i>35</i>
<i>Figura 2.10 – Resultado da tradução no Power Translator Pro.....</i>	<i>36</i>
<i>Figura 2.11 – O recurso de tradução interativa no Power Translator Pro.....</i>	<i>37</i>
<i>Figura 2.12 – Erros na tradução do texto pelo Power Translator.....</i>	<i>38</i>
<i>Figura 3.1 - Espiral do Conhecimento [Nonaka-Takeuchi 1997].....</i>	<i>50</i>
<i>Figura 3.2 - Como o ser humano identifica o domínio de um problema [Canuto 2000].....</i>	<i>52</i>
<i>Figura 3.3 -Estrutura de um sistema baseado em conhecimento [Canuto 2000].....</i>	<i>53</i>
<i>Figura 5.1 – Tela inicial do protótipo.....</i>	<i>69</i>
<i>Figura 5.2 – Tela de uso do sistema.....</i>	<i>70</i>
<i>Figura 5.3 – Informações obtidas de uma especificação TSL.....</i>	<i>72</i>

## **Lista de Tabelas**

<i>Tabela 2.1 – Análise da Tradução com o Babel Fish.....</i>	<i>30</i>
<i>Tabela 2.2 – Análise da Tradução com o Power Translator.....</i>	<i>39</i>

## Lista de Quadros

<i>Quadro 3.1 - Exemplo de uma entrada TMX</i> .....	49
<code>&lt;tu srclang="en-US" datatype="html" tuid="sample1"&gt;</code> .....	49
<code>&lt;tuv xml:lang="en" datatype="html"&gt;</code> .....	49
<code>&lt;seg&gt;link to &lt;bpt i="1" type="link" x="1"&gt;</code> .....	49
<code>&amp; a href="www.mysite.com" title="&lt;sub type="x-title"&gt;</code> .....	49
<code>my site&lt;/sub&gt;" &amp;gt;&lt;/bpt&gt;my web site&lt;ept i="1"&gt;&amp;lt;/a &amp;gt;, &lt;/ept&gt;</code> .....	49
<code>&lt;/seg&gt;</code> .....	49
<code>&lt;/tuv&gt;</code> .....	49
<code>&lt;tuv xml:lang="es" datatype="html"&gt;</code> .....	49
<code>&lt;seg&gt;enlace a &lt;bpt i="1" type="link" x="1"&gt;</code> .....	49
<code>&amp; a href="www.mysite.com/es" title="&lt;sub type="x-title"&gt;</code> .....	49
<code>mi sitio&lt;/sub&gt;" &amp;gt;&lt;/bpt&gt;mi sitio web&lt;ept i="1"&gt;&amp;lt;/a &amp;gt;, &lt;/ept&gt;</code> .....	49
<code>&lt;/seg&gt;</code> .....	49
<code>&lt;/tuv&gt;</code> .....	49
<code>&lt;/tu&gt;</code> .....	49
<i>Quadro 4.1 - Exemplo de uma Especificação TSL</i> .....	64
<code>&lt;TSL&gt;</code> .....	64
<code>&lt;HEAD&gt;</code> .....	64
<code>&lt;AUTHOR&gt;Marcello Novaes de Amorim&lt;/AUTHOR&gt;</code> .....	64
<code>&lt;CONTACT&gt;novaes@inf.ufes.br&lt;/CONTACT&gt;</code> .....	64
<code>&lt;DATE&gt;14-02-2007&lt;/DATE&gt;</code> .....	64
<code>&lt;DOM&gt;PORTUGUÊS-BR&lt;/DOM&gt;</code> .....	64
<code>&lt;DESC&gt;Tradutor de Idioma português para alguns outros idiomas&lt;/DESC&gt;</code> .....	64
<code>&lt;/HEAD&gt;</code> .....	64
<code>&lt;BODY&gt;</code> .....	64
<code>&lt;CASE&gt;</code> .....	64
<code>&lt;PATTERN&gt;PEGA LEVE&lt;/PATTERN&gt;</code> .....	64
<code>&lt;TA&gt;</code> .....	64
<code>&lt;DOM&gt;PORTUGUÊS FORMAL&lt;/DOM&gt;</code> .....	64
<code>&lt;PATTERN&gt;Tenha calma&lt;/PATTERN&gt;</code> .....	64
<code>&lt;SERVICE&gt;NEGRITO&lt;/SERVICE&gt;</code> .....	64

</TA>.....	64
<TA>.....	64
<DOM>INGLÊS COLOQUIAL</DOM>.....	64
<PATTERN>Take it easy</PATTERN>.....	64
<SERVICE>NULL</SERVICE>.....	64
</TA>.....	64
<TA>.....	64
<DOM>ESPAÑHOL COLOQUIAL</DOM>.....	64
<PATTERN>Coge ha conducido</PATTERN>.....	64
<SERVICE>DESTAQUE</SERVICE>.....	64
</TA>.....	64
</CASE>.....	64
.....	64
<b>Quadro 4.1 (continuação) - Exemplo de uma Especificação TSL</b> .....	<b>65</b>
.....	65
<CASE>.....	65
<PATTERN>[SUBSTANTIVO] [ADJETIVO]</PATTERN>.....	65
<TA>.....	65
<DOM>INGLÊS</DOM>.....	65
<PATTERN>[ADJETIVO] [SUBSTANTIVO]</PATTERN>.....	65
<SERVICE>NULL</SERVICE>.....	65
<CASE>.....	65
<PATTERN>ZÉ NINGUÉM</PATTERN>.....	65
<TA>.....	65
<DOM>INGLÊS COLOQUIAL</DOM>.....	65
<PATTERN>John Doe</PATTERN>.....	65
<SERVICE>DESTAQUE</SERVICE>.....	65
</TA>.....	65
<TA>.....	65
<DOM>PORTUGUÊS COLOQUIAL</DOM>.....	65
<PATTERN>Indigente</PATTERN>.....	65
<SERVICE>NEGRITO</SERVICE>.....	65
</TA>.....	65
<TA>.....	65
<DOM>INGLÊS FORMAL</DOM>.....	65

<b>&lt;PATTERN&gt;NULL&lt;/PATTERN&gt;</b> .....	65
<b>&lt;SERVICE&gt;NEGRITO&lt;/SERVICE&gt;</b> .....	65
<b>&lt;/TA&gt;</b> .....	65
<b>&lt;TA&gt;</b> .....	65
<b>&lt;DOM&gt;ESPAÑHOL&lt;/DOM&gt;</b> .....	65
<b>&lt;PATTERN&gt;NULL&lt;/PATTERN&gt;</b> .....	65
<b>&lt;SERVICE&gt;Dicionario_Portugues_Espanhol&lt;/SERVICE&gt;</b> .....	65
<b>&lt;/TA&gt;</b> .....	65
<b>&lt;TA&gt;</b> .....	65
<b>&lt;DOM&gt;ITALIANO&lt;/DOM&gt;</b> .....	65
<b>&lt;PATTERN&gt;NULL&lt;/PATTERN&gt;</b> .....	65
<b>&lt;SERVICE&gt;Dicionario_Portugues_Italiano&lt;/SERVICE&gt;</b> .....	65
<b>&lt;/TA&gt;</b> .....	65
<b>&lt;/CASE&gt;</b> .....	65
<b>&lt;/BODY&gt;</b> .....	65
<b>&lt;/TSL&gt;</b> .....	65
<b>Quadro 6.1 - Exemplo de Casos TSL</b> .....	82
<b>&lt;CASE&gt;</b> .....	82
<b>&lt;PATTERN&gt;PEGA LEVE&lt;/PATTERN&gt;</b> .....	82
<b>&lt;TA&gt;</b> .....	82
<b>&lt;DOM&gt;INGLÊS COLOQUIAL&lt;/DOM&gt;</b> .....	82
<b>&lt;PATTERN&gt;TAKE IT EASY&lt;/PATTERN&gt;</b> .....	82
<b>&lt;SERVICE&gt;NEGRITO&lt;/SERVICE&gt;</b> .....	82
<b>&lt;/TA&gt;</b> .....	82
<b>&lt;TA&gt;</b> .....	82
<b>&lt;DOM&gt;ESPAÑHOL COLOQUIAL&lt;/DOM&gt;</b> .....	82
<b>&lt;PATTERN&gt;COGE HA CONDUCIDO&lt;/PATTERN&gt;</b> .....	82
<b>&lt;SERVICE&gt;NEGRITO&lt;/SERVICE&gt;</b> .....	82
<b>&lt;/TA&gt;</b> .....	82
<b>&lt;/CASE&gt;</b> .....	82
<b>&lt;CASE&gt;</b> .....	82
<b>&lt;PATTERN&gt;[SUBSTANTIVO] [ADJETIVO]&lt;/PATTERN&gt;</b> .....	82
<b>&lt;TA&gt;</b> .....	82
<b>&lt;DOM&gt;INGLÊS COLOQUIAL&lt;/DOM&gt;</b> .....	82
<b>&lt;PATTERN&gt;[ADJETIVO] [SUBSTANTIVO]&lt;/PATTERN&gt;</b> .....	82



<SERVICE>DICIONARIO</SERVICE>.....	82
</TA>.....	82
<TA>.....	82
<DOM>ESPANHOL COLOQUIAL</DOM>.....	82
<PATTERN>NULL</PATTERN>.....	82
<SERVICE>DICIONARIO</SERVICE>.....	82
</TA>.....	82
</CASE>.....	82
.....	82
<b>Quadro 6.2 – Casos de ocultação de palavras proibidas.....</b>	<b>84</b>
<CASE>.....	84
<PATTERN>VAGABUNDO</PATTERN>.....	84
<TA>.....	84
<DOM>PORTUGUÊS COLOQUIAL</DOM>.....	84
<PATTERN>***</PATTERN>.....	84
<SERVICE>NOTIFICACAO</SERVICE>.....	84
</TA>.....	84
</CASE>.....	84
<CASE>.....	84
<PATTERN>BICHA</PATTERN>.....	84
<TA>.....	84
<DOM>PORTUGUÊS COLOQUIAL</DOM>.....	84
<PATTERN>***</PATTERN>.....	84
<SERVICE>NOTIFICACAO</SERVICE>.....	84
</TA>.....	84
</CASE>.....	84
<CASE>.....	84
<PATTERN>RETARDADO</PATTERN>.....	84
<TA>.....	84
<DOM>PORTUGUÊS COLOQUIAL</DOM>.....	84
<PATTERN>***</PATTERN>.....	84
<SERVICE>NOTIFICACAO</SERVICE>.....	84
</TA>.....	84
</CASE>.....	84
<CASE>.....	84

<b>&lt;PATTERN&gt;CORNO&lt;/PATTERN&gt;</b> .....	<b>84</b>
<b>&lt;TA&gt;</b> .....	<b>84</b>
<b>&lt;DOM&gt;PORTUGUÊS COLOQUIAL&lt;/DOM&gt;</b> .....	<b>84</b>
<b>&lt;PATTERN&gt;***&lt;/PATTERN&gt;</b> .....	<b>84</b>
<b>&lt;SERVICE&gt;NOTIFICACAO&lt;/SERVICE&gt;</b> .....	<b>84</b>
<b>&lt;/TA&gt;</b> .....	<b>84</b>
<b>&lt;/CASE&gt;</b> .....	<b>84</b>
.....	<b>84</b>

# SUMÁRIO

<b><u>1. Introdução.....</u></b>	<b><u>1</u></b>
<b>1.1. Contextualização.....</b>	<b>3</b>
<b>1.2. Objetivo.....</b>	<b>4</b>
<b>1.3. Justificativa.....</b>	<b>5</b>
<b>1.4. Metodologia.....</b>	<b>6</b>
<b><u>1.5. Estrutura da dissertação.....</u></b>	<b><u>8</u></b>
<b><u>2. Tradução.....</u></b>	<b><u>10</u></b>
<b><u>2.1. Tradução Automática.....</u></b>	<b><u>12</u></b>
<b><u>2.2. Estado da Arte.....</u></b>	<b><u>14</u></b>
<b><u>2.3. Paradigmas de Tradução Automática.....</u></b>	<b><u>15</u></b>
2.3.1. Paradigmas Fundamentais.....	15
2.3.1.1. Tradução Automática baseada em regras (RBMT).....	15
2.3.1.2. Tradução Automática baseada em conhecimento (KBMT).....	15
2.3.1.3. Tradução Automática baseada em léxico (LBMT).....	16
2.3.1.4. Tradução Automática baseada em restrições (CBMT).....	16
2.3.1.5. Tradução Automática baseada em princípios (PBMT).....	16
2.3.2. Paradigmas Empíricos.....	17
2.3.2.1. Tradução Automática baseada em estatística (SMT ou SBMT).....	17
2.3.2.2. Tradução Automática baseada em casos (EBMT).....	18
2.3.3. Paradigmas Híbridos.....	18
<b><u>2.4. Métodos de tradução automática.....</u></b>	<b><u>19</u></b>
2.4.1. Método Direto.....	20
2.4.2. Método por Transferência.....	21
2.4.3. Método Por Interlíngua.....	22
2.4.4. Memória de Tradução.....	23
2.4.5. Tradução dirigida por sintaxe.....	24
<b><u>2.5. Ferramentas para Tradução Automática.....</u></b>	<b><u>25</u></b>
2.5.1. Babel Fish.....	26
É possível observar que alguns dos erros encontrados poderiam ser evitados caso o Babel Fish contasse com um módulo de Memória de Tradução. Dentre esses erros os candidatos mais fortes a serem resolvidos por meio de Memória de Tradução são os pares: (look at, olhe) e (Baker Street, Rua Baker). Esse é um bom exemplo de como o uso conjunto de sistemas de tradução pode gerar uma tradução automática melhor.....	30
2.5.2. Power Translator Pro.....	31
<b><u>2.6. Um novo método.....</u></b>	<b><u>40</u></b>
2.6.1. Justificativa.....	41
2.6.2. Casos: Regras e Fatos.....	42
2.6.3. Processamento individual e independente dos padrões.....	43
2.6.4. Descrição do método.....	44
2.6.5. Novas Possibilidades.....	45
<b><u>3. Embasamento Teórico.....</u></b>	<b><u>46</u></b>
<b><u>3.1. Programação Declarativa.....</u></b>	<b><u>46</u></b>

<b>3.2. Linguagens de marcação.....</b>	<b>47</b>
<b>3.3. SGML.....</b>	<b>47</b>
<b>3.4. XML.....</b>	<b>48</b>
<b>3.5. O formato TMX.....</b>	<b>49</b>
<b>3.6. Conhecimento Tácito e Conhecimento Explícito.....</b>	<b>50</b>
<b>3.7. Bases de Conhecimento.....</b>	<b>51</b>
<b>3.8. Sistemas Baseados em Conhecimento.....</b>	<b>52</b>
<b>3.9. WebServices.....</b>	<b>54</b>
<i>Estão disponíveis gratuitamente no sítio do PHP, localizado no endereço <a href="http://www.php.net">www.php.net</a>, o manual completo da linguagem, exemplos, fóruns de discussão e todas as ferramentas necessárias para a construção e execução de sistemas PHP. ....</i>	<b>55</b>
<b>4. A TSL.....</b>	<b>56</b>
<b>4.1. O Conceito da Linguagem.....</b>	<b>56</b>
<b>4.2. A Gramática.....</b>	<b>57</b>
<b>4.3. A Estrutura.....</b>	<b>62</b>
4.3.1 Seção Head.....	62
4.3.2. Seção Body.....	62
4.3.3. Registro de Caso.....	62
4.3.4. Átomo de tradução.....	63
<b>4.4. Um exemplo de especificação TSL.....</b>	<b>64</b>
<b>5. Interpretadores TSL.....</b>	<b>66</b>
<b>5.1. Requisitos.....</b>	<b>66</b>
<b>5.2. Caso Padrão.....</b>	<b>67</b>
<b>5.3. O protótipo.....</b>	<b>68</b>
5.3.1. Obtendo os dados de uma especificação TSL.....	71
5.3.2. Acessando a gramática correta.....	73
5.3.3. Tradução sensível ao contexto.....	73
5.3.4. Acesso a aplicações externas.....	74
<b>6. Programas TSL.....</b>	<b>77</b>
<b>6.1. Os Conceitos.....</b>	<b>77</b>
<b>6.2. Como programar em TSL.....</b>	<b>78</b>
6.2.1. Definição do escopo.....	78
6.2.2. Elicitação do conhecimento do especialista.....	79
6.2.3. Determinar as ações do Caso Padrão.....	79
6.2.4. Formalização dos Casos.....	79
<b>6.3. Estudo de Caso.....</b>	<b>80</b>
Os serviços disponíveis para cada um dos exemplos serão diferentes, de acordo com o contexto do problema.....	80
6.3.1. Tradutor Multi-Idiomático.....	81
6.3.2. Eliminador de palavras proibidas.....	83
<b>7. Considerações Finais.....</b>	<b>85</b>
<b>7.1. Perspectivas futuras.....</b>	<b>86</b>
<b>8. Referências.....</b>	<b>87</b>



## 1. Introdução

Tudo é comunicação. O Sol comunica, por meio de ação gravitacional, aos planetas que devem permanecer em sua órbita, o vento comunica, por meio de forças mecânicas, às árvores que devem se balançar, um leão comunica a todos, por meio de rugidos, que é o macho dominante de um dado território, e parceiros de poker combinam jogadas por meio de sinais discretos como olhares e movimentos de dedos.

A comunicação torna possível o dinamismo do universo, as entidades não podem agir isoladas, pois fazem parte de um sistema, e dessa forma, devem interagir com as outras partes do conjunto para que tudo flua de forma harmônica.

A questão fundamental da comunicação é: A mensagem enviada deve ser compreendida da forma como o remetente pretendeu. Dessa questão surgem os problemas básicos, os problemas de comunicação. Para que uma mensagem seja compreendida é preciso que o destinatário entenda, e tenha condições de assimilá-la e para tanto, é necessário que quem a receba tenha domínio da linguagem na qual a mesma se encontra, e que tenha uma carga de conhecimento acumulado que permita uma inferência das partes implícitas nesta mensagem.

Considerando isso, é possível afirmar que quanto mais claras estiverem as informações, menor esforço cognitivo será necessário para sua compreensão, e uma vez que a inferência realizada por indivíduos diferentes pode não ser a mesma, uma mensagem clara e direta tem mais chances de ser compreendida corretamente, ou seja, da forma como o autor pretendeu.

Existem vários fatores que dificultam o entendimento, em especial é possível ressaltar dois deles:

- Diversas linguagens podem ser utilizadas para representar idéias, e é possível que o destinatário não compreenda aquela escolhida pelo remetente.
- O remetente e o destinatário podem ter níveis diferentes de conhecimento do domínio em questão.

Ao dissecar uma mensagem, observa-se sua composição, a qual é formada por conceitos, que de modo geral, dizem respeito aos objetos e assuntos tratados, e relacionamentos, estes representam as relações e interações entre eles.

Conseqüências e informações adicionais podem ser inferidas a partir do conhecimento armazenado relativo ao universo associado ao contexto da mensagem, porém não se pode garantir que sejam compreendidos os conceitos e relacionamentos que não estão explícitos.

### **1.1. Contextualização**

Um processo de tradução geralmente é usado para criar representações que permitam (ou facilitem) o entendimento do conjunto de conceitos e relacionamentos explícitos em uma mensagem. A qualidade da tradução está associada à semelhança entre o conteúdo semântico do modelo gerado em relação ao conteúdo no original.

O computador é uma poderosa ferramenta, pois permite a descrição e especificação de métodos capazes de generalizar e automatizar o processo.

A tradução automática existe, porém, em relação à linguagem natural, ela encontra uma série de problemas decorrentes das ambigüidades, e outras características inerentes à mesma, como o uso de metáforas, ironia e várias exceções que requerem um tratamento diferenciado para a preservação da integridade semântica.

Existem abordagens, como a memória de tradução, que visa preservar a semântica de segmentos encontrados com certa freqüência, e outras como a tradução por sintaxe, que prima por processar grandes blocos sem a interação humana, deixando que uma futura revisão corrija as incoerências semânticas. A questão importante é determinar qual o método adequado para situação; em muitos casos, a utilização conjunta de diferentes métodos fornece um resultado final bem interessante.

Há também a tradução semi-automática, realizada por sistemas que permitem interação humana durante o processo.



## 1.2. Objetivo

O objetivo é propor uma linguagem, capaz de representar o conhecimento necessário para realizar transformações em um texto, a partir da identificação de padrões e a correspondente execução de ações programadas.

A TSL fornece mecanismos que permitem a identificação de padrões em mensagens e torna possível o processamento individual e independente de cada um deles. Isso permite que sejam executadas múltiplas tarefas durante o processamento, seguindo uma especificação TSL.

O conceito da linguagem em foco permite a organização dos elementos que formam cada padrão e o seu envio reorganizado, caso seja desejado um processamento adicional, para diferentes funções, módulos, WebServices ou qualquer outro tipo de sistema computacional com o qual seja capaz de se comunicar.

Resumindo, a proposta é fornecer uma linguagem declarativa baseada na identificação e tratamento de padrões, disparando ações programadas quando estes forem identificados e permitindo recorrer a sistemas externos para aplicá-las.

### **1.3. Justificativa**

Uma linguagem que permite explicitar o conhecimento de especialistas, de uma maneira que pode ser facilmente processada por um computador, cria um universo de possibilidades. Permite a pessoas que não dominam técnicas de programação, contribuir para a criação de sistemas computacionais de diversos tipos.

O suporte à delegação de tarefas para outros sistemas e a capacidade de integrar as traduções parciais de cada padrão, possibilita a geração de documentos multimídia, traduções multi-idiomáticas e diversos outros tipos de saídas que requerem formas variadas de processamento.

A geração automática desses documentos de saída, eliminando a necessidade de do uso seqüencial ou simultâneo de vários programas, proporciona maior produtividade e conforto ao usuário.

#### 1.4. Metodologia

O método adotado para o desenvolvimento do trabalho, foi um misto entre pesquisa e prototipagem.

Inicialmente, a idéia era adaptar o SGMT [Novaes 2006], um sistema gerenciador de Memórias de Tradução, para transformá-lo em um Sistema de Produção. O propósito principal era criar um sistema capaz de gerar documentos multimídia a partir do processamento de textos, um produtor multimídia.

Como o sistema inicial apenas tratava Fatos, ou seja, padrões fixos de texto, nas primeiras experiências o produtor multimídia agia basicamente como uma memória de tradução, identificando os Fatos, e substituindo por objetos pré-determinados. Esses objetos já podiam ser imagens, vídeos, sons, textos (com ou sem formatação), enfim, era uma memória de tradução que fornecia varias possibilidades para substituir partes do texto de entrada.

Em paralelo, estava sendo desenvolvida uma pesquisa para identificar métodos que produzissem traduções automáticas de qualidade satisfatória, quando utilizados de forma complementar à Memória de Tradução.

Essa pesquisa chamou atenção para a utilização conjunta da Memória de Tradução e um método baseado em regras gramaticais (Regras). Esse método de tradução identifica padrões em um texto, que tenham sido previstos em regras gramaticais armazenadas em sua base de dados. Foi uma variante da Tradução Dirigida por Sintaxe.

Os resultados motivaram a implementar no protótipo a funcionalidade de reconhecer regras gramaticais. Nesse momento, o produtor multimídia era capaz de reconhecer Fatos e Regras em um texto, e substituir os padrões encontrados.

Conforme os testes eram realizados, com Fatos e Regras armazenados em banco de dados, surgiu a idéia de flexibilizar o sistema, permitindo uma tradução para mais de um idioma. Os registros que armazenavam as regras e os fatos guardavam informações sobre como os padrões identificados deveriam ser substituídos, de acordo com o idioma-destino desejado. Isso permitiu parametrizar o sistema, ou seja, definir para qual idioma a tradução deve ser realizada.

Levantamentos bibliográficos revelaram o TMX [LISA 2007], um formato que permite a representação de memórias de tradução, com os padrões equivalentes em vários idiomas para cada segmento de texto. Uma característica que chamou atenção no TMX foi ele ser baseado na XML, o que tornava extremamente fácil sua

criação e facilitava seu compartilhamento, processamento e integração com sistemas computacionais.

O TMX não atendia aos requisitos para criar especificações capazes de descrever traduções multimídia baseadas no tratamento de Regras e Fatos, mas serviu de inspiração para a criação de uma nova linguagem que suprisse essas necessidades.

Após uma análise das bases de dados que eram utilizadas no protótipo, e da forma como os registros eram aplicados no processamento, foi definida uma estrutura baseada em linguagens de marcação. Era o primeiro esboço da TSL (Translator Specification Language), que até então não tinha nome, mas já tinha o propósito de descrever especificações de tradutores com suporte multimídia.

No protótipo, poucas linhas de código diferenciavam os métodos que especificavam as Regras dos que especificavam os Fatos. No decorrer dos testes, percebeu-se que o processamento dos mesmos era muito similar, o que revelou a possibilidade de criar uma generalização que contemplasse Regras e Fatos, a essa generalização deu-se o nome de Caso.

A partir desse avanço, a TSL se tornou bem mais simples, compacta, de mais fácil legibilidade e processamento e assim foi criada a versão atual da linguagem, que é o tema dessa dissertação. O protótipo, que inicialmente era o foco do trabalho, será apresentado na seção 5.3.

## 1.5. Estrutura da dissertação

Esta dissertação está estruturada obedece à estrutura descrita nessa seção. Existe uma parte inicial, que conta com o sumário, lista de tabelas, dicionário de dados, agradecimentos e folhas de rosto.

O capítulo 1 é introdutório, e tem como objetivo contextualizar o leitor no universo de assuntos tratados no documento. O capítulo 8 apresenta uma lista das referências utilizadas durante o processo de pesquisa e escrita, e como embasamento teórico para afirmações feitas ao longo do texto.

O conteúdo principal da dissertação está dividido nos capítulos de 2 a 7, cada um deles com um objetivo específico.

- **Capítulo 2: Tradução**

Este capítulo aborda vários aspectos do tema “Tradução”, apresenta conceitos, descreve seus paradigmas e métodos, e propõe um novo método de tradução.

- **Capítulo 3: Embasamento Teórico**

O capítulo de embasamento teórico tem como objetivo esclarecer o leitor, sobre todos os conceitos e tecnologias que foram utilizadas no decorrer do processo de produção, e que serão citados ao longo dos capítulos posteriores.

- **Capítulo 4: A Linguagem TSL**

O capítulo 4 apresenta a linguagem TSL, que é o foco principal desse trabalho. São apresentados seus conceitos, sua gramática e um exemplo de especificação escrita nessa linguagem.

- **Capítulo 5: Interpretadores TSL**

São levantados os requisitos necessários para que um ambiente seja reconhecido como um interpretador TSL, feitas propostas implementações e é apresentado como estudo de caso, o interpretador utilizado nos experimentos.

- **Capítulo 6: Aplicações TSL**

O capítulo 6 explica como é possível utilizar a TSL para descrever sistemas computacionais e apresenta um exemplo de aplicação descrita como uma especificação TSL.

- **Capítulo 7: Considerações finais**

O capítulo de considerações finais relata as observações feitas ao longo dos testes, pesquisa e desenvolvimento. Também são feitas propostas para que no futuro o uso da TSL seja viável em diferentes ambientes e para que ela seja amplamente utilizada.

## 2. Tradução

O dicionário Aurélio [Aurélio 1988] dá três definições para tradução:

1. Transpor, transladar, duma língua para outra.
2. Explicar, manifestar.
3. Simbolizar.

O Houaiss [Houaiss 2003] propõe mais definições:

1. Operação que consiste em fazer passar um enunciado emitido numa determinada língua (língua-fonte) para o equivalente em outra língua (língua-alvo), ambas conhecidas pelo tradutor; assim, o termo ou discurso original torna-se compreensível para alguém que desconhece a língua de origem.
2. Transposição de uma mensagem de uma forma gráfica para outra.
3. Sentido figurado.
4. Aquilo que reflete, que expressa de modo indireto; repercussão, imagem, reflexo.

Foi realizada uma pesquisa, com indivíduos de diferentes idades e escolaridades, com a pergunta: "O que é tradução?". As respostas foram totalmente diversas, mas foi possível tirar algumas conclusões sobre o que o termo "Tradução" significa para as pessoas.

A maior parte das pessoas entrevistadas acredita que tradução é transcrever um texto, em um idioma diferente do original. Porém, foi observado que as pessoas com maior grau de instrução, que participam de atividades coletivas ou tem maior acesso à informação, associaram tradução à representação de informações de uma forma que as torne acessível a quem às deseja acessar.

Essas duas linhas, embora semelhantes em sua essência, demonstram claramente com qual finalidade a tradução é utilizada por diferentes grupos. Outro ponto que se pode inferir a partir das respostas obtidas, é que conforme as pessoas tem acesso à informação, outras representações, além da escrita, são utilizadas no processo de comunicação, e por esse motivo, são aptas a serem traduzidas, e a forma como são representadas, são aceitas como representação resultante de uma tradução.

Antes de se chegar a uma definição formal do que é tradução, é necessário entender o que é traduzir. Entender quais são as partes do objeto a ser traduzido e o que acontece a elas durante o processo de tradução.

Uma análise prática mostra que um objeto que se pode traduzir seja um texto ou qualquer outra forma de informação, é formado basicamente de conceitos e relacionamento entre eles. Essa estrutura, composta de conceitos e relacionamentos resulta em um conteúdo que tem uma carga semântica. Essa carga semântica é o que deve ser preservado durante o processo de tradução.

Dessa forma, é possível determinar que o requisito básico de uma tradução é representar os conceitos contidos no objeto original, preservando o seu conteúdo. Assim a informação contida no objeto traduzido será equivalente à informação contida no original.

A carga semântica está fortemente associada aos relacionamentos entre os conceitos, e por este motivo devem representados de alguma maneira, mesmo que não exista forma de representar no domínio-destino alguns dos conceitos relacionados.

A partir disso é possível definir tradução como a representação de um conjunto de conceitos e relacionamentos, em uma forma diferente da original, preservando o conteúdo semântico original.



## 2.1. Tradução Automática

A Tradução Automática é uma das aplicações mais antigas da computação e se mostra mais importante na medida que se estabelece o cenário globalizado, pois a variedade de línguas é uma barreira para a comunicação e compartilhamento de informação.

A possibilidade de automação da tradução humana vem sendo discutida desde o século XVII, mas foi durante a partir de 1949, com Guerra Fria, que Americanos e Soviéticos crentes na viabilidade de um tradutor totalmente automatizado de alto desempenho e qualidade, deram os primeiros passos relevantes no desenvolvimento desse tipo de sistema.

O primeiro congresso sobre tradução automática ocorreu em 1952, no MIT, quando foi apresentado um relatório que reconhecia que os sistemas de tradução automática poderiam ser desenhados para cumprir diferentes expectativas: tradução de alta qualidade nos domínios da ciência, das finanças e da diplomacia, por exemplo; e tradução rápida, talvez de baixa exatidão, para a varredura de textos de inimigos em jornais, revistas e panfletos. O principal obstáculo identificado foi a ambigüidade semântica, identificado foi como ponto limitante da qualidade e não como impedimento. Nessa época surgiram os primeiros modelos híbridos de tradução automática ("mixed MT"), com pré-edição e pós-edição, mas seu desempenho foi bastante discutido por conta da grande preocupação da época com a velocidade das traduções.

Esse evento merece destaque, assim como o relatório apresentado que antecipou muito dos desdobramentos posteriores sobre o tema além de chamar atenção à necessidade da utilização de dicionários e analisadores gramaticais como componentes dos sistemas. O número de pesquisadores interessados na ciência da tradução automática aumentou consideravelmente resultando em maior número de publicações, experimentos e simulações, e conseqüentemente avanços na área.

A primeira crítica sistemática sobre os sistemas de tradução foi emitida em um relatório por Bar-Hillel em 1960, apontando a impossibilidade de se desenvolver um sistema de tradução de alta qualidade completamente automático, alegando que todo conhecimento humano deveria estar disponível para que um sistema fosse capaz de realizar a tarefa satisfatoriamente. Sugeriu que as pesquisas fossem voltadas aos sistemas de tradução automática assistida por humanos.

O Comitê Assessor de Processamento Automático das línguas Naturais, da Academia de Ciências dos Estados Unidos, encarregado da análise dos resultados dos programas subsidiados pelo governo Norte-Americano, emitiu em 1966 o relatório ALPAC criticando os resultados alcançados e questionando a utilidade da tradução automática. Isso provocou um impacto negativo afastando por no cerca de 20 anos os investimentos e o número de pesquisadores interessados.

No início dos anos 80 a tradução automática voltou a ganhar força, principalmente na Europa e Japão, com outros objetivos menos ambiciosos, pois a primeira fase havia mostrado que o processo somente seria viável se a complexidade da tarefa pudesse ser reduzida.

## 2.2. Estado da Arte

Atualmente o termo “Tradução Automática” se refere a uma variedade de práticas de pesquisa e projetos de desenvolvimento, muitos dos quais fogem do objetivo inicial de automatizar completamente o processo de tradução de idiomas.

Diante da aparente inviabilidade da meta original, alguns desenvolvedores redefiniram progressivamente o foco para sistemas de tradução assistidos por humanos, outros decidiram operar em domínios específicos, abrangendo linguagens mais formais. Diante desta situação, surgiram duas visões de tradução:

- Tradução Grosseira ou Rudimentar: Utilizada como ferramenta para que o usuário possa tomar a decisão de requisitar ou não uma tradução humana. Seu uso em sistemas de busca de informações e aplicações do gênero.
- Tradução Crua: Planejada para a pós-edição, produzindo resultados parciais e imperfeitos que servem como de ponto de partida para um tradutor humano.

### **2.3. Paradigmas de Tradução Automática**

Os paradigmas de tradução referem-se aos componentes de representação de conhecimento que auxiliam o projeto de processamento global dos sistemas de tradução.

#### **2.3.1. Paradigmas Fundamentais**

São aqueles que empregam teorias lingüísticas bem definidas, utilizando restrições sintáticas, lexicais ou semânticas, sobre as línguas naturais envolvidas. Estes paradigmas podem ser subdivididos nos grupos a seguir.

##### **2.3.1.1. Tradução Automática baseada em regras (RBMT)**

São caracterizados por representar o conhecimento por meio de regras de diferentes níveis lingüísticos, para a tradução entre as línguas fonte e alvo. Por exemplo, para a transferência lexical, as características e restrições de itens lexicais individuais são codificadas num mecanismo de controle, por meio de regras, e não no léxico.

##### **2.3.1.2. Tradução Automática baseada em conhecimento (KBMT)**

Define sistemas baseados em regras que utilizam conhecimentos profundos, lingüísticos ou extralingüísticos, de um domínio, permitindo que o sistema possa tecer inferências sobre os conceitos manipulados. Segundo Kay a maior justificativa para utilização de sistemas KBMT é que a tradução depende fortemente de informações e características extralingüísticas, de senso comum e de conhecimento do mundo. A representação do conhecimento pode envolver o desenvolvimento de ontologias e modelos de domínio [Kay 1994].

### **2.3.1.3. Tradução Automática baseada em léxico (LBMT)**

São aqueles que fornecem regras para relacionar as entradas lexicais de uma língua às entradas lexicais de outra língua. O mapeamento é realizado por meio de um léxico bilíngüe que associa diretamente árvores fonte e alvo por meio de ligações entre itens lexicais e seus argumentos. De modo simplificado, cada entrada nesse léxico bilíngüe contém regras para o mapeamento entre a sentença na LF e a sentença na LA.

### **2.3.1.4. Tradução Automática baseada em restrições (CBMT)**

Permite definir restrições em vários níveis de descrição lingüística, por exemplo, para os itens lexicais. Nesse paradigma, as operações de mapeamento requeridas na transferência são executadas por equações de transferência baseadas em restrições que relacionam estruturas dos textos fonte e alvo.

### **2.3.1.5. Tradução Automática baseada em princípios (PBMT)**

Sistemas PBMT (Principle-Based Machine Translation) são uma alternativa aos sistemas RBMT, nos quais as regras são substituídas por um pequeno conjunto de princípios que envolvem fenômenos morfológicos, gramaticais e lexicais, de um modo geral. Um exemplo de construção derivada de princípios gerais é a construção da voz passiva. Como não existe uma única regra de mapeamento entre duas línguas naturais para a voz passiva, é comum utilizar-se um conjunto de princípios que definem as operações morfológicas e sintáticas necessárias.

O paradigma PBMT é complementar às abordagens KBMT e EBMT, no sentido de que ele provê uma cobertura ampla para muitos fenômenos lingüísticos, mas lhe falta conhecimento mais profundo sobre o domínio de tradução.

### **2.3.2. Paradigmas Empíricos**

São aqueles que utilizam pouca ou nenhuma teoria lingüística no processo de tradução. Em geral, eles indicam técnicas experimentais para especificar o mecanismo de tradução apropriado ao contexto em foco.

#### **2.3.2.1. Tradução Automática baseada em estatística (SMT ou SBMT)**

Utilizam técnicas estatísticas ou probabilísticas que contemplam as tarefas lingüístico-computacionais em foco na tradução (por exemplo, a desambigüização lexical). A idéia dessa abordagem é que a tradução seja realizada por meio de dados estatísticos extraídos automaticamente de corpora de textos bilíngües paralelos.

Alguns exemplos de dados que podem ser obtidos a partir da análise desses corpora são:

- Probabilidade de uma sentença fonte ocorrer no texto-alvo;
- Probabilidade de uma palavra fonte ser traduzida como uma ou mais palavras alvo;
- Probabilidade de tradução de cada palavra em outra palavra da língua alvo;
- Probabilidade da posição de cada palavra na sentença na língua fonte, quando essa posição não é a mesma que a da palavra na sentença alvo.

### **2.3.2.2. Tradução Automática baseada em casos (EBMT)**

Em vez de regras de mapeamento entre as línguas, utiliza-se um procedimento que tenta combinar o texto a ser traduzido com exemplos de traduções armazenados. A tradução é, portanto, por analogia com exemplos coletados a partir de traduções já realizadas, os quais são anotados com suas descrições superficiais, em um corpus bilíngües alinhado.

Basicamente, a idéia é utilizar um algoritmo de unificação para encontrar o exemplo mais próximo da sentença de entrada, a partir do corpus bilíngües. Esse procedimento resulta num modelo de tradução, o qual pode, então, ser preenchido palavra-por-palavra, de acordo com as palavras da sentença de entrada. A proximidade de cada exemplo com a sentença de entrada é determinada pela distância semântica entre as suas palavras, a qual pode ser calculada com base na distância entre essas palavras em uma hierarquia de termos e conceitos provida, em geral, por um thesaurus ou uma ontologia.

A combinação de frases requer pelo menos uma análise sintática básica das traduções paralelas, além de alguma análise semântica para determinar a proximidade da combinação. Assim, a tradução de sentenças exige também que a estrutura sintática da sentença fonte seja combinada com sentenças no corpus. A maioria dos sistemas EBMT não considera a combinação da sentença inteira, mas sim de algumas de suas partes, como sintagmas nominais ou preposicionais.

### **2.3.3. Paradigmas Híbridos**

Muitos paradigmas, principalmente os empíricos, apresentam dificuldades para manipular alguns aspectos do processo de tradução automática. Assim, é reconhecida a necessidade de combinar paradigmas de forma a explorar as vantagens de cada um.

## 2.4. Métodos de tradução automática

Dois tipos de informação podem ser utilizados para classificar um sistema de tradução automática; seu método e seu paradigma. Os métodos se referem ao projeto de processamento, ou seja, à organização global do processamento de seus vários componentes. Enquanto a seção 2.3 trata dos paradigmas, esta escreve os métodos mais utilizados.

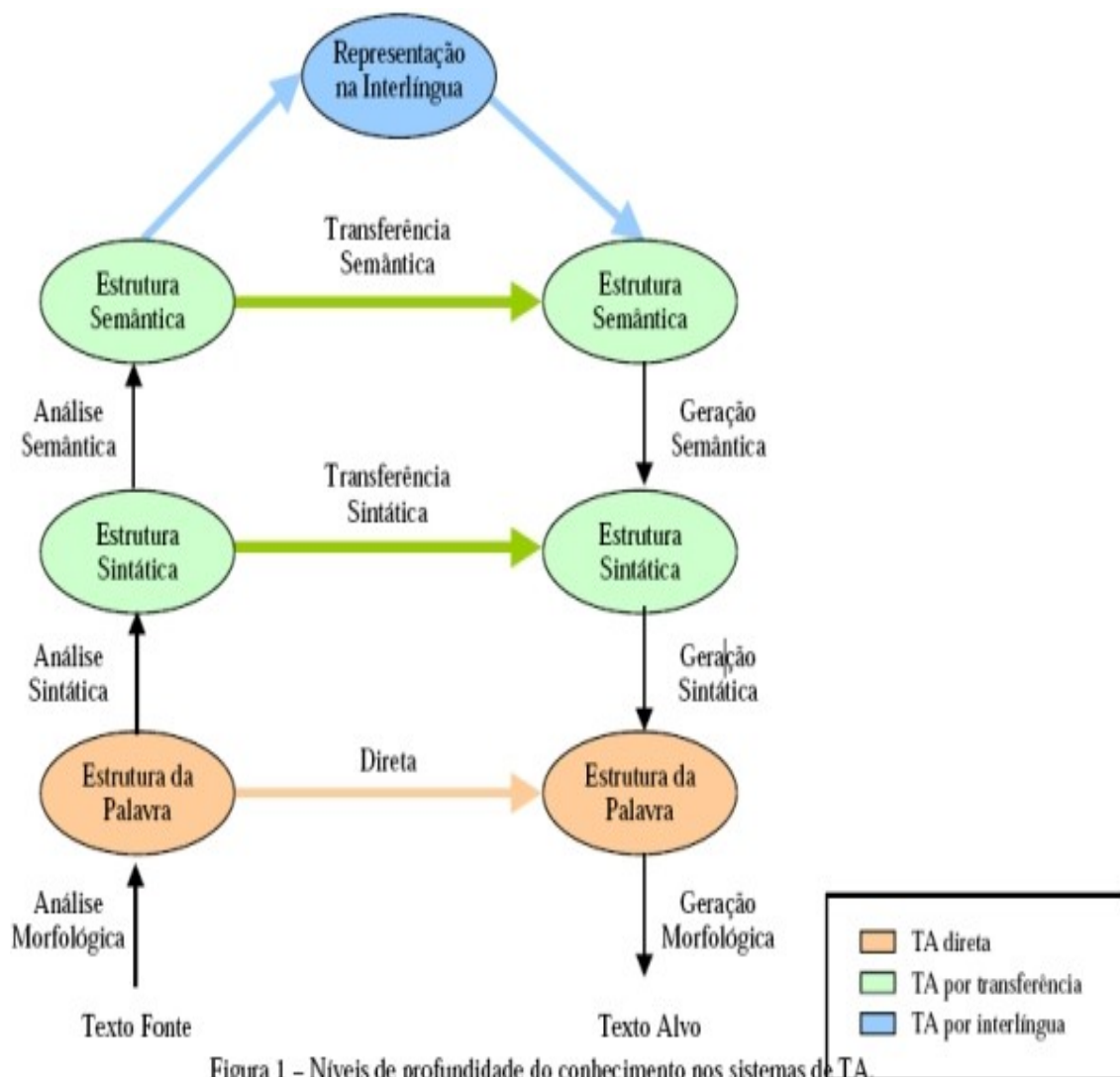


Figura 1 - Níveis de profundidade do conhecimento nos sistemas de TA.

Figura 2.1 - Relacionamento entre os diversos métodos de tradução



### **2.4.1. Método Direto**

A tradução automática direta transforma as sentenças da LF em sentenças da LA sem utilizar representações intermediárias, procurando realizar o mínimo de processamento lingüístico possível. Esse processamento pode variar, incluindo a simples substituição das palavras de uma sentença-fonte por sua(s) correspondente(s) na LA (tradução palavra-por-palavra) ou a realização de tarefas mais complexas, como a reordenação das palavras na sentença-alvo e a inclusão de preposições.

Este método apresenta uma complexidade relativamente baixa, e utiliza necessariamente um dicionário bilíngüe, uma gráfica, e regras de transformação da linguagem fonte para linguagem alvo.

#### **Componentes Necessários:**

- Dicionários bilíngües;
- Um conjunto de regras de transformação da língua fonte para a alvo.

### 2.4.2. Método por Transferência

Na tradução automática por transferência, a tradução consiste nos passos:

1. Alteração da estrutura e palavras da sentença de entrada resultando em uma representação intermediária da LF (fase de análise);
2. Transformação dessa representação em uma estrutura intermediária da LA (fase de transferência);
3. Geração da sentença na LA (fase de geração), a partir dessa estrutura.

#### Componentes Necessários:

- Gramáticas e léxicos substanciais de ambas as línguas, os quais são utilizados tanto na análise das sentenças fonte, quanto na geração das sentenças alvo;
- Dicionários bilíngües para as regras de substituição de palavras;
- Uma gramática comparativa, ou seja, um conjunto de regras de transformação para relacionar a representação intermediária da LF com a representação intermediária da LA;

A fase de análise pode envolver processos complexos como as análises semântica e gramática, mas, em geral, são mais comuns sistemas que se limitam à análise sintática, gerando como representação intermediária uma estrutura de árvore. Nesse caso, a fase de transferência converte essa estrutura da LF em uma estrutura de árvore da LA, por meio de regras de mapeamento entre as duas línguas naturais, que indicam as correspondências lexicais e sintáticas entre tais estruturas. Para tanto, é necessário representar o conhecimento contrastivo (i.e., comparativo) das duas línguas, o qual envolve a especificação de suas diferenças normativas e lexicais.

A fase de geração transforma a estrutura de árvore da LA na sentença final, propriamente dita, utilizando a gramática e o léxico da LA.

### 2.4.3. Método Por Interlíngua

Devido à dificuldade de se estabelecer regras de transferência e recursos lingüísticos comparativos (como gramáticas) efetivos, necessários aos sistemas desenvolvidos sob o método de tradução automática por transferência, e também à complexidade inerente a esses sistemas (seção 2.4.2.), houve o interesse pela definição de um nível de análise tão profundo a ponto de permitir descartar os componentes contrastivos entre as línguas em foco, presentes na tradução por transferência. O objetivo era fazer com que a saída da análise da LF correspondesse diretamente à entrada do componente de geração na LA.

Representações nesse nível deveriam capturar, assim, o significado a ser transmitido, independentemente da língua natural em questão. Esta é justamente a função da Interlíngua: permitir extrair a representação do significado da sentença fonte para, a partir dela, gerar a sentença na LA.

#### **Componentes Necessários:**

- Gramáticas e léxicos substanciais de ambas as línguas, os quais são utilizados tanto na análise das sentenças fonte, quanto na geração das sentenças alvo;
- Dicionários bilíngües para as regras de substituição de palavras;
- Um conjunto de regras de transformação para relacionar a Interlíngua com as línguas fonte e alvo.

#### **2.4.4. Memória de Tradução**

A Memória de Tradução é uma técnica que visa preservar a semântica de padrões que são inerentes de uma linguagem, como expressões idiomáticas, provérbios e metáforas, ou seja, memória de tradução consiste em armazenar fatos (frases e expressões), assim como suas traduções pré-determinada, de forma a permitir a substituição desses padrões sem alterar o sentido original [Breda 2005].

Por suas características, a Memória de Tradução é sensível ao contexto, podendo ser implementada de forma a relacionar representações equivalentes em diversas linguagens diferentes para um dado padrão.

Uma Memória de Tradução precisa ser construída, alimentada durante sua utilização. Dessa forma, quanto mais se usa uma Memória de Tradução, mais completa ela se torna. Outra tendência desse tipo de sistema, quando utilizado de forma individual, é se tornar cada vez mais personalizado, refletindo o conhecimento de seu utilizador nas traduções realizadas.

Outra característica das Memórias de Tradução, é que assim como os dicionários, elas podem ser compartilhadas. Existem maneiras específicas de se compartilhar Memórias de tradução, dentre elas, o TMX (Translation Memory Exchange), que é um formato baseado nas linguagens de marcação, que tem como finalidade armazenar uma Memória de Tradução em um arquivo que possa facilmente ser compartilhado.

O uso da Memória de Tradução ganhou força nos anos 90, quando a demanda por tradutores automáticos aumentou por conta do comércio internacional. Surgiram projetos com o objetivo de durante as traduções, coletar as expressões mais usadas e suas traduções para construir bancos de expressões para Memórias de Tradução [Garrão 2001].

#### **2.4.5. Tradução dirigida por sintaxe**

A tradução dirigida por sintaxe utiliza regras gramaticais para encontrar padrões em um texto, ao encontrar um desses padrões é aplicada uma ação semântica seguindo regras de produção.

Sendo utilizada em conjunto com dicionários, essa abordagem torna possível a tradução integral de um texto, de forma automática. Porém, como é baseada em regras gramaticais a TDS não é capaz de preservar a semântica de expressões idiomáticas, termos técnicos etc. Dessa forma, é possível entender as vantagens de utilizar uma Memória de Tradução e a tradução dirigida por sintaxe de forma conjunta.

## **2.5. Ferramentas para Tradução Automática**

Existem hoje diversos sistemas de tradução de textos. Cada um deles com características positivas e negativas, mas de modo geral apresentam o mesmo problema para o usuário: muitas vezes as correções sobre a tradução gerada automaticamente é mais trabalhosa que a tradução manual.

Uma utilização conjunta dos sistemas existentes, aplicando cada um deles aos casos em que apresenta melhores resultados, pode ser uma solução interessante para melhorar a qualidade da tradução gerada automaticamente. Porém é extremamente trabalhoso e muito pouco prático trabalhar com vários programas de tradução ao mesmo tempo.

Na seção 2.6 é apresentado um novo método que propõe algo inspirado nessa idéia de delegação de processamento para múltiplos sistemas de tradução (ou outros tipos de sistema), e no capítulo 4 é apresentada uma proposta para que essa idéia possa ser implementada.

Antes, porém, é necessário apresentar nessa seção alguns sistemas de tradução amplamente utilizados e analisar as traduções geradas, para que se possa ilustrar, em um modelo real, os problemas citados referentes à tradução automática.

### 2.5.1. Babel Fish

O Babel Fish é um sistema para a tradução de textos, que teve seu nome inspirado em uma criatura homônima apresentada no livro de Douglas Adams, O guia do Mochileiro das Galáxias [Guia 1979]. Ao ser inserido no ouvido, o Babel Fish (Peixe Babel) permite ao seu hospedeiro compreender textos verbalizados em qualquer idioma do universo.

Definição no idioma original [Guia 1979]:

*“The Babel fish is small, yellow and leechlike, and probably the oddest thing in the Universe. It feeds on brainwave energy received not from its own carrier but from those around it. It absorbs all unconscious mental frequencies from this brainwave energy to nourish itself with. It then excretes into the mind of its carrier a telepathic matrix formed by combining the conscious thought frequencies with nerve signals picked up from the speech centers of the brain which has supplied them. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language.”*

Tradução para o português:

*“O peixe-babel é pequeno, amarelo e semelhante a uma sanguessuga, e é provavelmente a criatura mais estranha em todo o Universo. Alimenta-se de energia mental, não daquele que o hospeda, mas das criaturas ao redor do hospedeiro. Absorve todas as frequências mentais inconscientes desta energia mental e se alimenta delas, e depois expele na mente de seu hospedeiro uma matriz telepática formada pela combinação das frequências mentais conscientes com os impulsos nervosos captados dos centros cerebrais responsáveis pela fala do cérebro que os emitiu. Na prática, o efeito disto é o seguinte: se você introduz no ouvido um peixe-babel, você compreende imediatamente tudo o que lhe for dito em qualquer língua.”*

O sistema de tradução Babel Fish pode ser acessado gratuitamente, por meio de uma interface WEB acessível no endereço <http://babelfish.altavista.com> .

Como pode ser observado na figura 2.2, a interface WEB do Babel Fish é bem simples, permitindo a digitação de um texto ou a especificação de uma página de Internet a ser traduzido.



The image shows the Babel Fish translation website interface. At the top, the title "Tradução Babel Fish" is displayed in white text on a blue background, accompanied by a small yellow fish icon. To the right, there is a link for "Ajuda". Below the title, the main heading is "Traduza um bloco de texto". Underneath this heading is a large, empty white text input area. Below the input area, there is a small instruction: "Use o [Teclado Mundial](#) para digitar caracteres acentuados ou em cirílico." Below this instruction, there is a dropdown menu labeled "Selecione os idiomas de e para" and a button labeled "Traduzir".

Below the first section, the heading is "Traduza uma página da Web". Underneath this heading is a white text input area containing the text "http://". Below the input area, there is a dropdown menu labeled "Selecione os idiomas de e para" and a button labeled "Traduzir".

At the bottom of the interface, there is a section titled "Adicione [Traduções Babel Fish](#) ao seu site." followed by a tip: "Dica: É possível agora seguir os links nas páginas de Web traduzidas." To the right of this text is a logo that says "POWERED BY SYSTRAN".

Figura 2.2 – Interface WEB do Babel Fish

A interface é intuitiva, tornando sua utilização fácil, mesmo para pessoas com pouca familiaridade com sistemas de tradução e/ou aplicações on-line.



O idioma origem e destino são escolhidos por meio de listas, conforme é possível observar na figura 2.3 .




Figura 2.3 - Selecionando os idiomas de e para no Babel Fish

Após escolhido o par, e digitado o texto ou link da página da web, basta clicarmos no botão traduzir.

Faltam opções de tradução direta entre idiomas, como por exemplo, do Espanhol para o Português. Porém isso pode ser obtido utilizando o sistema em etapas, mas de modo geral, os resultados obtidos dessa forma não são satisfatórios.

O Babel Fish não é configurável, nem é possível inserir novas traduções. Existem palavras que não estão contidas em sua base de tradução, e no caso de ocorrerem em um texto, o sistema mantém a palavra original, para que o usuário traduza da forma mais adequada. .

A figura 2.4 apresenta um exemplo no qual o texto não foi integralmente traduzido, pois o Babel Fish não conhece equivalência no idioma destino para algumas das palavras ou formas sentenciais que aparecem no texto original.

**Tradução Babel Fish**  [Ajuda](#)

**Em português:**

Na altura desta história, eu estava vivendo ainda meu no amigo Sherlock Holmes' flat na rua do padeiro em Londres. Muito cedo uma manhã, uma mulher nova, vestida no preto, veio ver-nos. Olhou tired e infeliz, e sua cara era muito branca. "eu estou receoso. Receoso da morte, Sr. Holmes!" gritou. "ajude-me por favor! Eu não sou trinta ainda e não olho meu cabelo cinzento! Eu estou assim receoso!"

**Traduza novamente**

Street in London. Very early one morning, a young woman, dressed in black, came to see us. She looked tired and unhappy, and her face was very white. "I'm afraid. Afraid of death, Mr. Holmes!" she cried. "Please help me! I'm not thirty yet and look at my grey hair! I'm so afraid!"

Use o [Teclado Mundial](#) para digitar caracteres acentuados ou em cirílico.

Inglês para português

**Adicione [Traduções Babel Fish](#) ao seu site.**  
**Dica:** Se não deseja que uma palavra seja traduzida adicione um x em cada um dos lados. Por exemplo: I love xPinkx xFloydx


POWERED BY 

Figura 2.4 – Texto exemplo traduzido no Babel Fish

Destacadas com retângulos estão as palavras que não puderam ser traduzidas, e sublinhados estão os segmentos que não foram traduzidos corretamente, ou seja, tiveram sua semântica alterada. Esses dois problemas são muito comuns nas traduções automáticas.

Na tabela 2.1 são analisados os erros detectados no exemplo da figura 2.4. É uma quantidade grande de erros em relação ao tamanho do texto traduzido, e somada a essa quantidade de erros, ainda houve os problemas de palavras não traduzidas por limitações dos dicionários disponíveis para o sistema.

<b>Inglês</b>	<b>Babel Fish</b>	<b>Correta</b>
at my friend Sherlock	meu no amigo Sherlock	no apartamento do meu
Holmes' flat	Holmes' flat	amigo Sherlock Holmes
Baker Street	rua do padeiro	Baker Street
dressed in black	vestida em negro	vestida de preto
I'm not thirty	Eu não sou trinta	Eu não tenho trinta anos
look at	não olho	olhe
so	assim	tão

**Tabela 2.1 – Análise da Tradução com o Babel Fish**

É possível observar que alguns dos erros encontrados poderiam ser evitados caso o Babel Fish contasse com um módulo de Memória de Tradução. Dentre esses erros os candidatos mais fortes a serem resolvidos por meio de Memória de Tradução são os pares: (look at, olhe) e (Baker Street, Rua Baker). Esse é um bom exemplo de como o uso conjunto de sistemas de tradução pode gerar uma tradução automática melhor.

### 2.5.2. Power Translator Pro

O Power Translator Pro é um sistema de tradução pago, criado pela empresa Globalink. Ele conta com dicionários de Inglês, Espanhol, Português, Francês e Italiano, oferecendo opções de traduzir do Inglês para outro desses idiomas, ou de algum dos outros idiomas para o Inglês.

Novamente, para realizar uma tradução entre dois idiomas diferentes do Inglês, é necessário que se utilize o programa repetidamente, aumentando as chances de erros decorrentes da tradução automática.

Esses erros acumulativos conseqüentes de realimentação do sistema com a saída da tradução anterior podem deteriorar tanto a semântica do texto a ponto de transformar o texto original em um amontoado de segmentos sem nexos.

A figura 2.5 apresenta a tela inicial do sistema, na qual o usuário escolhe os idiomas origem e destino da tradução.



Figura 2.5 – Selecionando Origem/Destino no Power Translator

Será utilizado o mesmo fragmento de uma das histórias de Sherlock Holmes que foi utilizado no exemplo da figura 2.4. O idioma Origem será o Inglês e o Destino será o Português.

No Power Translator Pro é possível acompanhar os passos da tradução. É possível observar na figura 2.6, que inicialmente o texto na janela inferior (janela de saída) é o mesmo que aparece na superior (texto original).

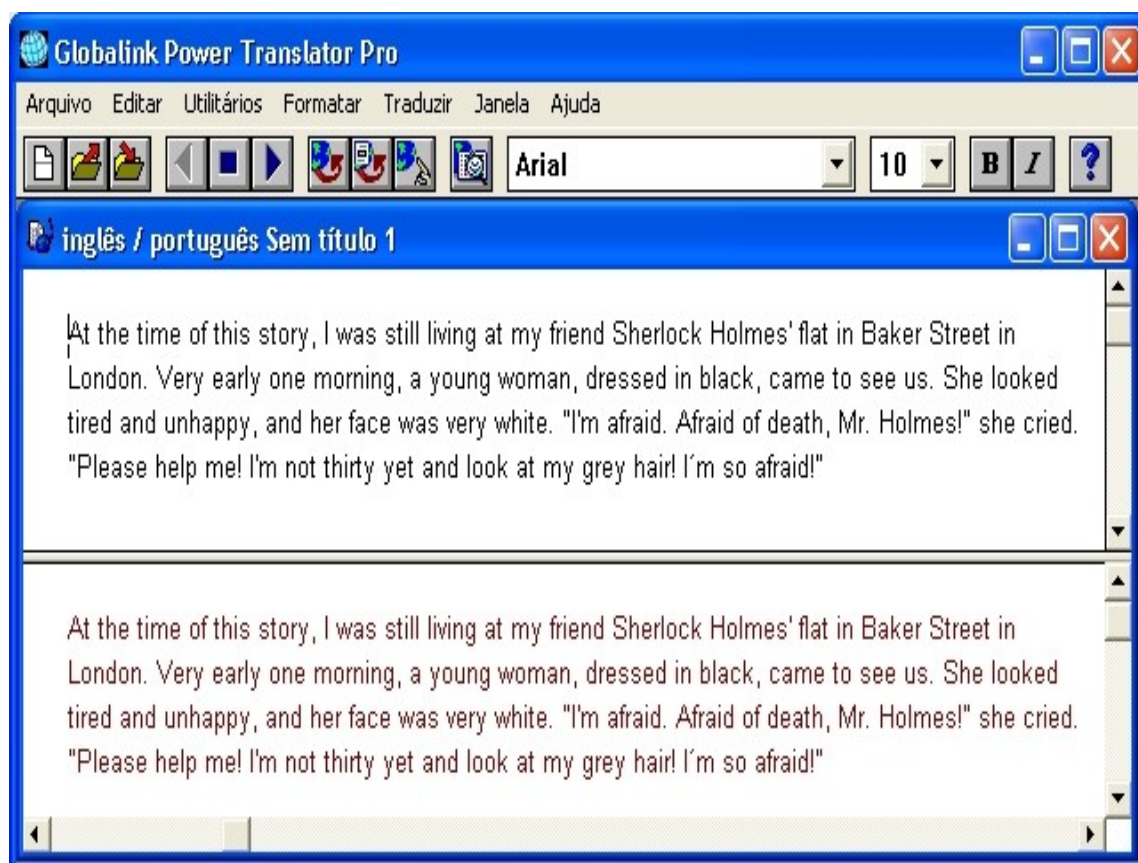


Figura 2.6 – Entrando com um texto no Power Translator

O texto é segmentado em orações, que são delimitadas pela pontuação. É possível navegar entre as frases por meio dos botões na barra de ferramentas com os ícones representando setas à esquerda e à direita. Para selecionar a oração existe o botão com o ícone em forma de quadrado, similar ao “stop” encontrado nos aparelhos eletrônicos e programas multimídia.

Na figura 2.7 uma frase é selecionada, e automaticamente em torno do segmento equivalente na janela de saída, aparece uma moldura, indicando que os dois segmentos de texto estão associados.

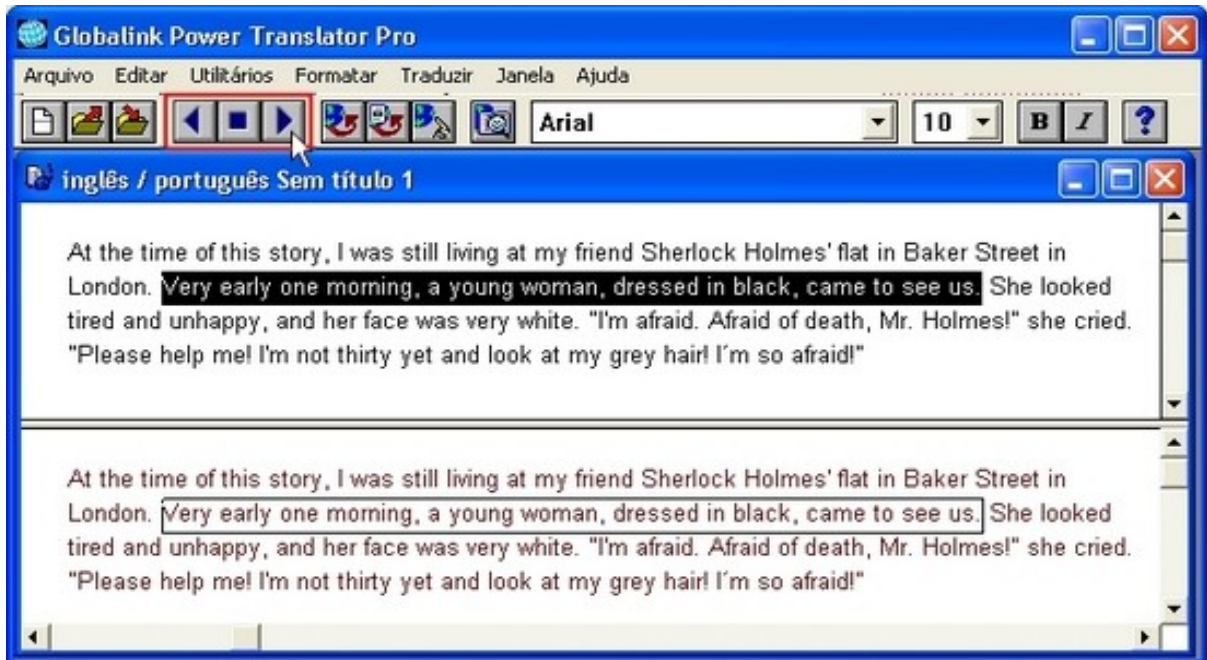


Figura 2.7 – Selecionando uma frase no Power Translator

Esse recurso é muito útil para que o usuário esteja sempre ciente de onde as ações estão sendo aplicadas. É possível executar a ação de tradução somente na região selecionada, o que permite uma correção incremental dos erros de tradução automática, evitando que eles se acumulem.

A figura 2.8 mostra a mesma seleção de orações da figura 2.7, porém é possível observar que foram aplicadas ações de tradução sobre o segmento selecionado, e o resultado substituído na janela de saída. Para isso, foi utilizado o botão “Traduzir Oração”, que age somente sobre a oração selecionada.

Mesmo após a tradução a relação de associação entre os dois fragmentos de texto se mantém, isso pode ser verificado pela persistência da seleção em ambas as janelas.

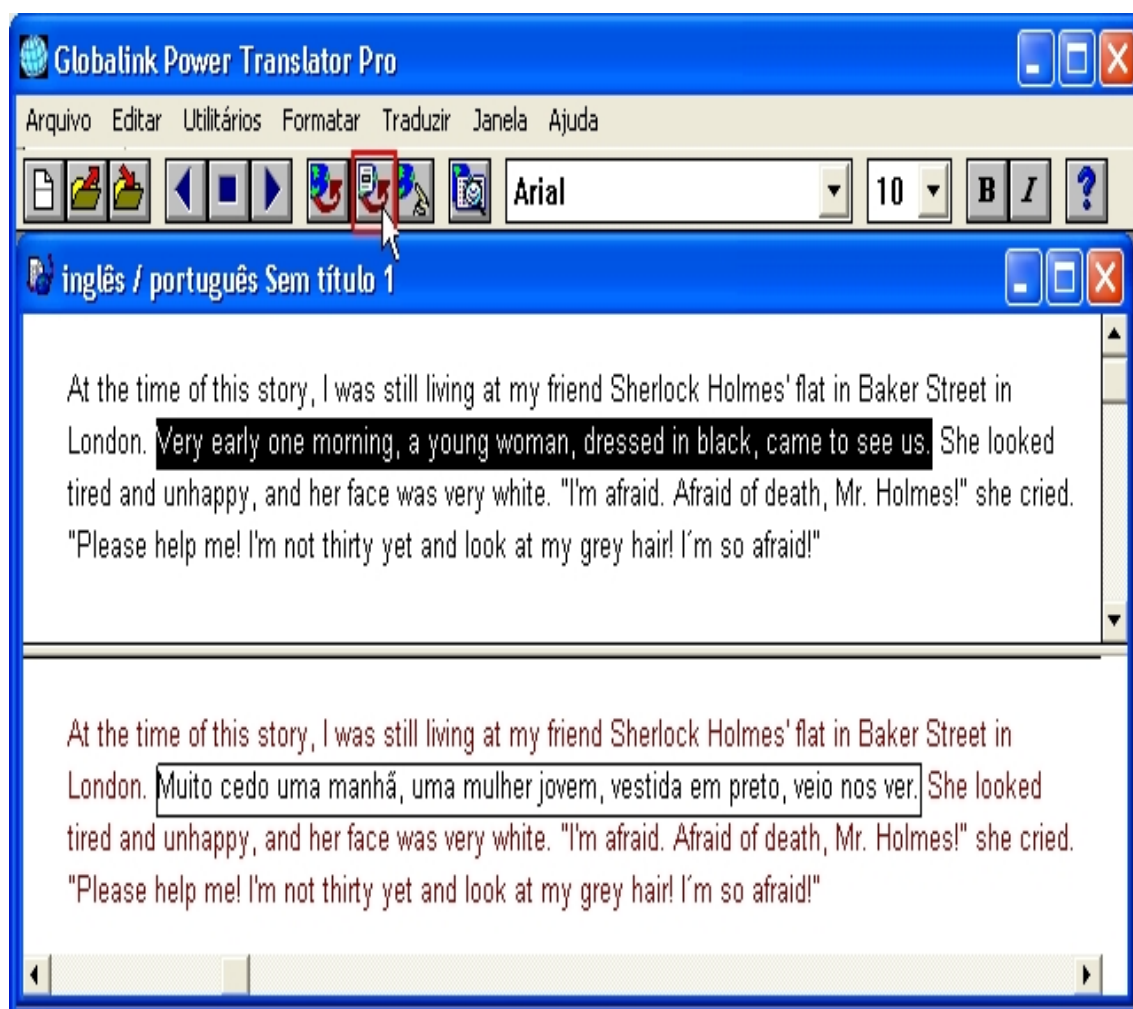


Figura 2.8 – Traduzindo uma frase no Power Translator



O botão "Traduzir Documento" aplica ações de tradução em todas as orações do texto, seqüencialmente. Dessa forma é possível utilizar o botão "Para" ou "Pausa" quando se deseja interromper a tradução. Nesse caso, as orações já traduzidas, tem as suas novas representações mantidas na janela de saída, mas o programa não continua a traduzir as orações restantes.

Uma barra de progresso informa qual a oração traduzida a cada momento e, com a opção "Atualizar Visualizador de Projeto" habilitada, é possível acompanhar em tempo real os passos da tradução, e caso seja observado algum erro, existe a possibilidade de interromper a tradução, e corrigir manualmente antes de prosseguir.

A figura 2.9 mostra as ações de tradução sendo executadas seqüencialmente nas orações do texto

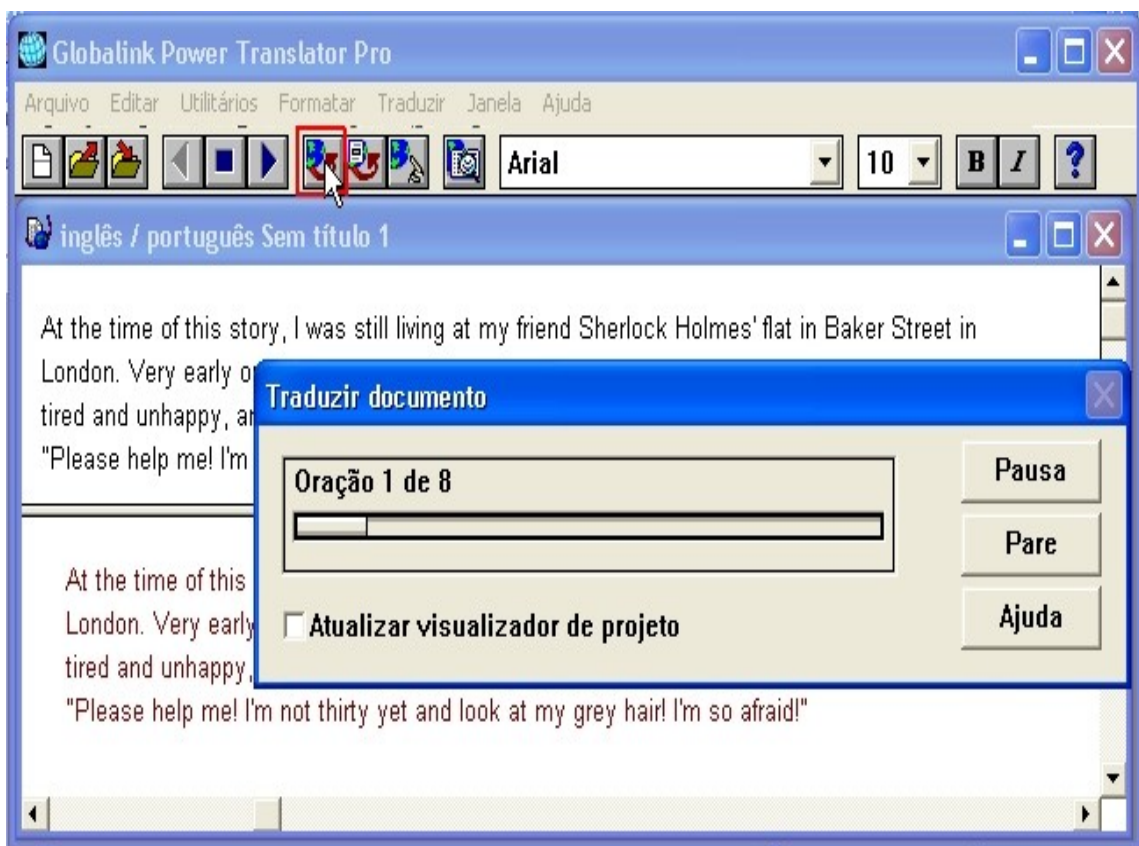


Figura 2.9 – Traduzindo um texto no Power Translator



O resultado da tradução realizada na figura 2.9 pode ser observado na figura 2.10.

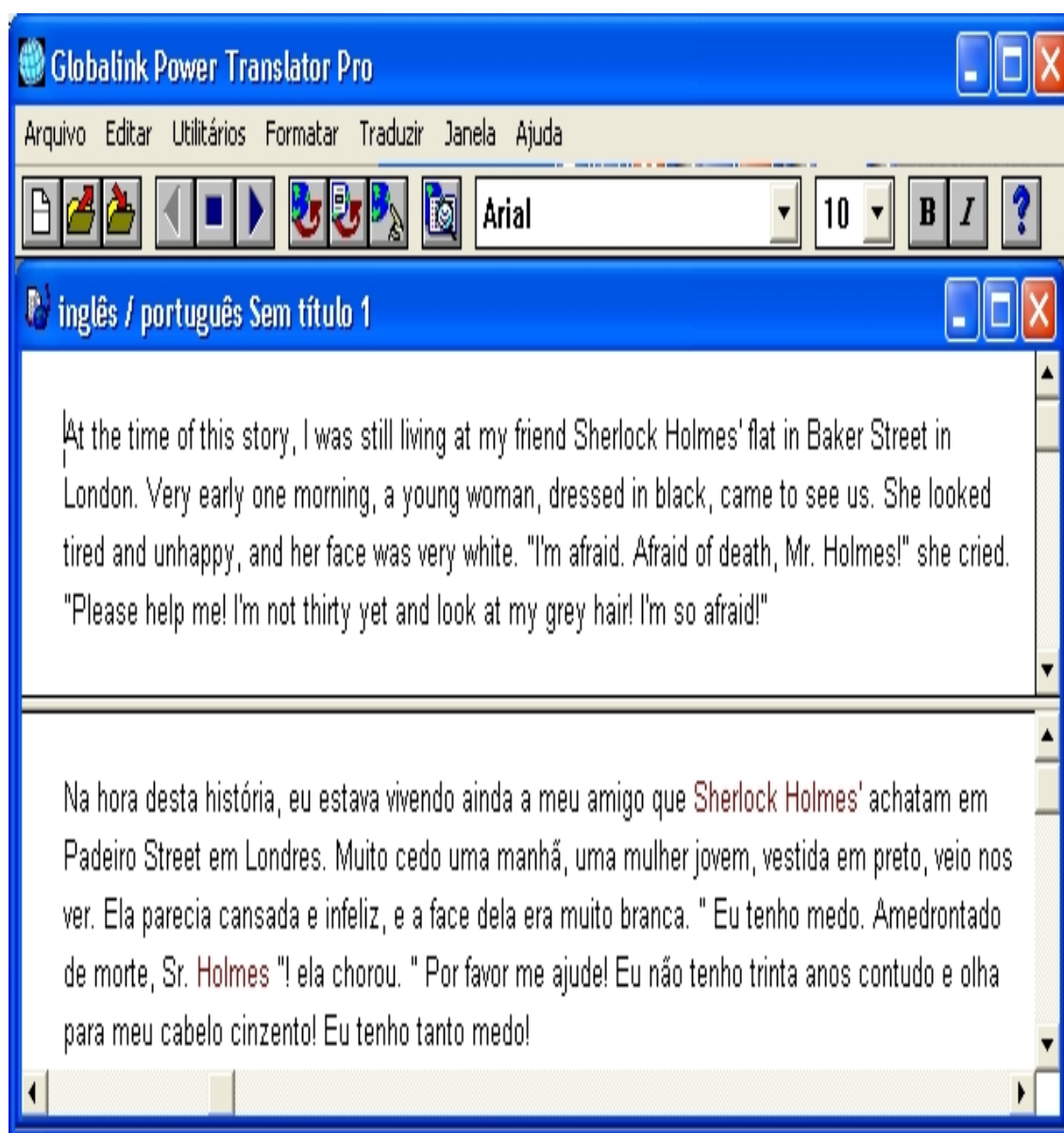


Figura 2.10 – Resultado da tradução no Power Translator Pro

Uma vez concluída a tradução, o Power Translator Pro pode-se utilizar o modo interativo para corrigir manualmente os termos das orações. Novamente são utilizados os botões de navegação para selecionar a oração desejada.

Observa-se na figura 2.11 que para cada uma das palavras contidas na oração selecionada são exibidos o seu equivalente no texto original e todas opções de tradução disponíveis, ou seja, inferidas pelo sistema.

Bastando selecionar as opções desejadas com o mouse, e apertar no botão “OK” ao concluir as alterações, para que a oração seja salva, e a nova tradução exibida na janela de saída na posição referente àquela oração.

A figura 2.11 mostra a interação do usuário. O sistema havia traduzido automaticamente a palavra “flat” como “achata” e o usuário corrigiu para “apartamento”.

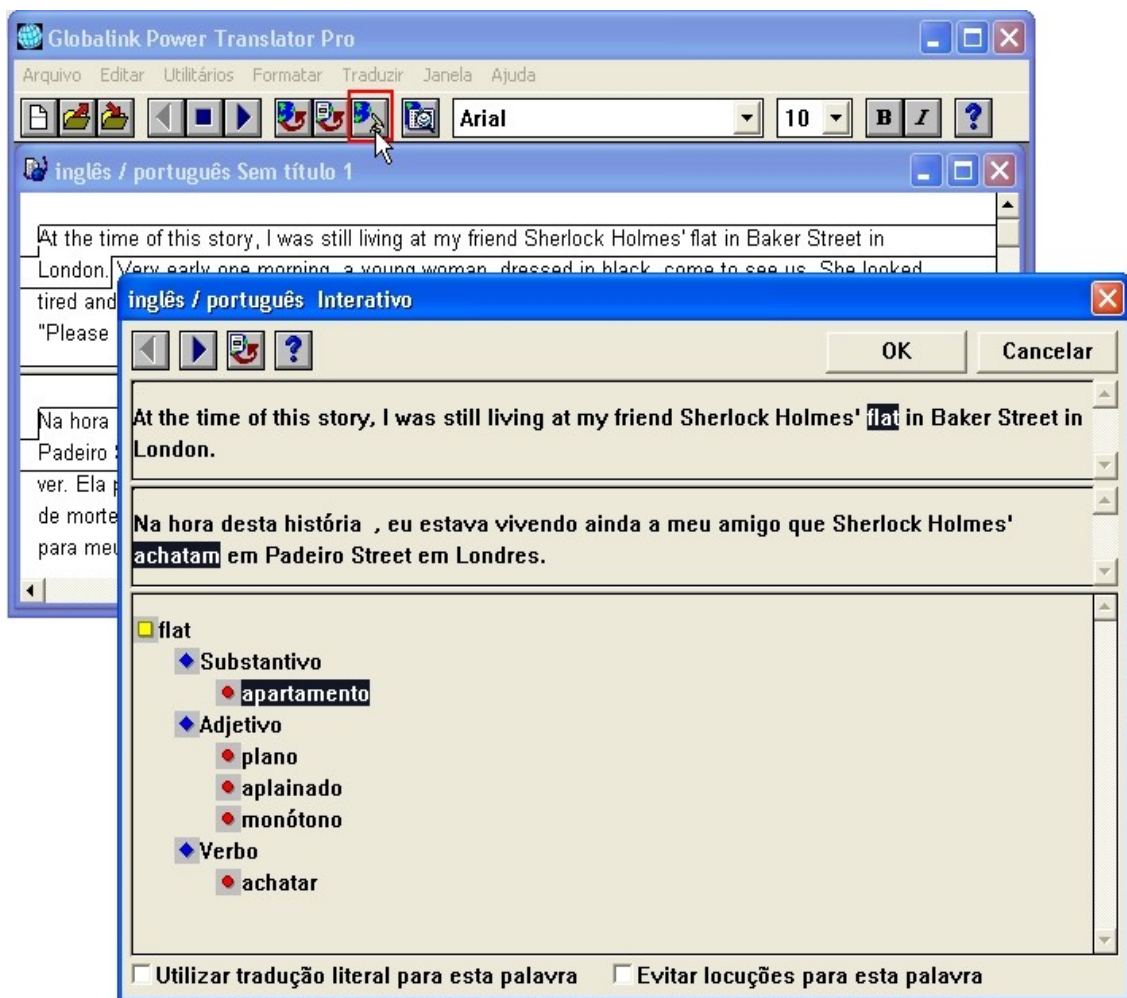


Figura 2.11 – O recurso de tradução interativa no Power Translator Pro

A figura 2.12 exibe o resultado da tradução após a interação do usuário, ilustrada na figura 2.11.

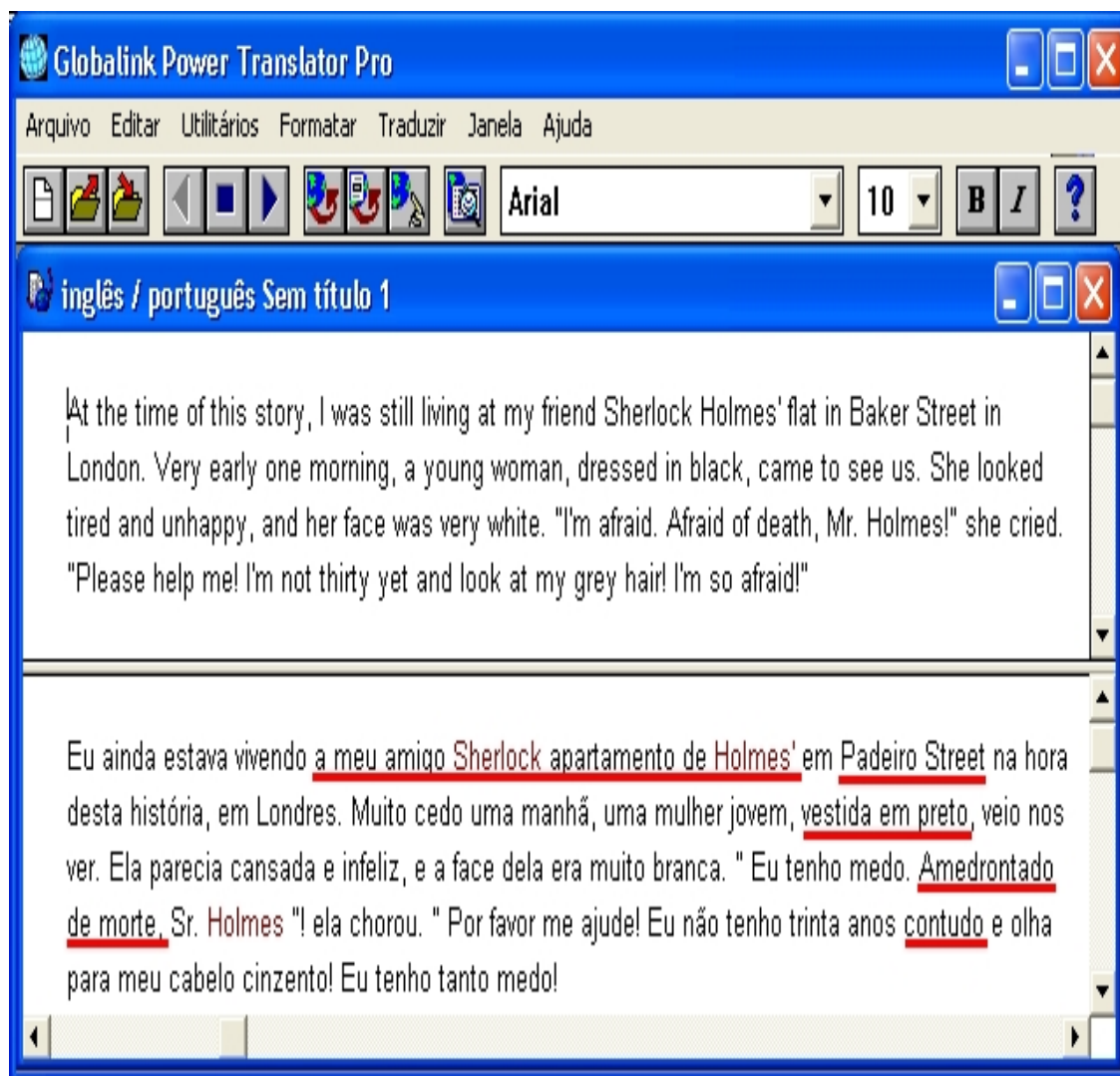


Figura 2.12 – Erros na tradução do texto pelo Power Translator

Em vermelho estão sublinhados os erros que persistiram na tradução resultante. Nenhuma palavra deixou de ser traduzida por não estar contemplada no dicionário, porém, erros comuns às traduções automáticas apareceram no texto traduzido.

Na tabela 2.2 é feita uma análise da tradução gerada com o auxílio do Power Translator Pro:

<b>Inglês</b>	<b>Power Translator</b>	<b>Tradução Correta</b>
at my friend Sherlock Holmes' flat	a meu amigo Sherlock apartamento de Holmes'	no apartamento do meu amigo Sherlock Holmes
Baker Street	Padeiro Street	<i>Baker Street</i>
dressed in black	vestida em negro	vestida de preto
Afraid of death	Amedrontado de morte	Medo da morte
yet	contudo (conjunção)	anida (advérbio)
<b>Tabela 2.2 – Análise da Tradução com o Power Translator</b>		

Utilizando o modo interativo do sistema, seria possível melhorar a qualidade da tradução gerada, porém, fornecer ferramentas para que o usuário corrija os erros da tradução automática não diminui significativamente a carga de trabalho necessária para corrigir a tradução. Mesmo com o uso dessas ferramentas, o número de erros cometidos na tradução automática pode ser muito grande para que seja viável utilizar o sistema em alguns casos.

O Power Translator Pro permite que sejam adicionadas novas entradas nos dicionários, inclusive adicionando expressões, o que fornece uma Memória de Tradução ao sistema. O uso desse recurso pode melhorar em muito a qualidade das traduções conforme novas entradas são adicionadas, mas isso pode demorar um longo tempo, e de modo geral, os usuários desistem de utilizar o sistema antes de o sistema começar a apresentar bons resultados.

## 2.6. Um novo método

1. Seja  $M$  o conjunto de todos os modelos mentais válidos no domínio  $D$ .
2. Seja  $R$  o conjunto de todas as regras que são capazes de gerar modelos mentais válidos em  $D$ , e  $M_r$  o conjunto de todos os modelos mentais válidos que podem ser gerados a partir de  $R$ .
3. Considerando 1 e 2, existe um conjunto  $F$ , tal que  $F$  é o complemento de  $M_r$  em relação à  $M$ . Dessa forma, é possível afirmar que  $F$  é um conjunto relacionado a exceções, ou seja, é um conjunto de fatos que representam modelos mentais válidos que não poderiam ser gerados por regras pertencentes à  $R$ .

Essas premissas mostram a possibilidade de determinar um conjunto de regras e exceções (fatos) capaz de processar um texto integralmente, gerando a partir da assimilação da tradução produzida, modelos mentais válidos que refletem a carga semântica contida na mensagem original, ou seja, é possível determinar uma lista de regras e fatos que especificam um mapeamento semântico entre domínios/representações.

O método proposto utiliza essa lista para identificar padrões em uma mensagem, e utiliza uma variante da Memória de Tradução para definir as ações semânticas que devem ser aplicadas sobre cada padrão gerar a tradução desejada. O processamento de cada padrão é independente, com um conjunto próprio de regras, o que cria um universo de possibilidades quanto ao resultado final.

Como a base de conhecimento (Regras e Fatos) utilizada nessa abordagem é gerada como Memória de Tradução, conforme é alimentada pelo usuário, a tradução automática se aproxima da gerada manualmente por ele.

### **2.6.1. Justificativa**

O uso das memórias de tradução comprovou o quanto é eficiente utilizar o conhecimento de tradutores humanos no processamento de fatos (padrões fixos) encontrados nas mensagens preservando a semântica, mas de modo geral esses padrões representam uma pequena percentagem da mensagem, sendo necessário o uso de outros métodos para concluir a tradução.

Os métodos de tradução dirigidos por sintaxe, são muito eficientes ao traduzir grandes blocos utilizando regras gramaticais, mas o resultado gerado normalmente não é semanticamente satisfatório sendo necessária uma correção manualmente posterior.

Adicionando a um sistema de Memória de Tradução a capacidade de além reconhecer e aplicar regras gramaticais é possível que o profissional da área de tradução especifique as regras que utiliza assim como as exceções (fatos). Assim, os grandes blocos de texto serão processados pelas regras gramaticais, e as exceções reconhecidas e tratadas de acordo com os fatos, aliando o desempenho à preservação da integridade semântica.

A possibilidade de tratar individualmente e de maneira diferente cada um dos padrões identificados, seja uma regra ou um fato, permite que seja utilizado uma função ou um programa diferente para processar cada um deles, gerando qualquer tipo de saída desejada a partir do conhecimento acumulado.

### 2.6.2.Casos: Regras e Fatos

O método proposto se baseia em que, após algum tempo de uso e alimentação, uma base de conhecimento pode conter entradas suficientes para reconhecer padrões em toda uma mensagem, a segmentando integralmente em uma lista de formada por eles.

- Seja  $C$  uma mensagem, formada pelos elementos  $C(i)$ .
- Seja " $i$ " é um número natural que representa a posição da mensagem na qual o elemento se encontra.
- Seja  $P$  um conjunto de agrupamentos  $P(j)$ .
- Seja " $j$ " sendo um numero natural que indica o índice do subconjunto.
- Seja  $P(j)$  um subconjunto de  $C$ , formado por 1 ou mais elementos de  $C$ .
- Um elemento  $C(i)$  pode pertencer a apenas um agrupamento  $P(j)$ .
- Em uma situação ideal, a união de todos os agrupamentos  $P(j)$  deve gerar um conjunto equivalente a  $C$ .
- Cada agrupamento  $P(j)$  pode ser entendido por um padrão, que pode ser tanto um Fato quanto uma Regra.
- Um Fato é um padrão fixo que é representado na base de conhecimento exatamente como aparece nas mensagens.
- Uma Regra é uma generalização, é o equivalente a uma regra gramatical da qual podem derivar diversas formações, inclusive algumas que podem estar representadas na mesma base de conhecimento como um Fato.
- Ao se atribuir um peso maior a um Fato do que a uma Regra, é definida uma relação de regra e exceção, ou seja, os Fatos podem funcionar como exceções às Regras.

O conjunto de Fatos tem uma função equivalente à Memória de Tradução, sendo adequados para representar expressões idiomáticas, provérbios populares e outras formações do gênero. De modo geral, devem ser representados como Fatos, todos os padrões sobre os quais não se pode preservar a semântica traduzindo de forma algorítmica.

Com o objetivo de criar uma generalização, foi criada a nomenclatura Caso que pode ser um Fato ou uma Regra, sendo associado a um registro que permite reconhecer um padrão e determinar a forma como ele deve ser processado.

### **2.6.3. Processamento individual e independente dos padrões**

É possível traduzir um texto dividindo-o em uma seqüência de padrões, e aplicando um diferente conjunto de ações sobre cada um deles.

Lembrando, que cada registro da base de conhecimento armazena um Caso capaz de reconhecer ao menos um padrão e especificar as ações que devem ser aplicadas para que seja gerada uma tradução parcial referente a cada padrão identificado.

Pode-se assim representar a maneira como um tradutor humano trata cada Caso, registrando as sutilezas da tradução e as particularidades que diferenciam uma tradução automática direta, de uma tradução elaborada feita por um profissional. Cada registro corretamente criado e adicionado à base de conhecimento, pode ser acessado para gerar uma saída semelhante ao produto de uma tradução realizada pelo seu autor sobre o padrão reconhecido pelo Caso.



#### **2.6.4. Descrição do método**

O método é simples, composto de três passos:

1. Consiste em percorrer a mensagem desde o início, procurando por padrões que se "encaixem" em um dos Casos, lembrando que em caso de haver uma Regra e um Fato que possam ser associados ao padrão, o Fato tem prioridade. O primeiro passo tem como saída uma lista de padrões.
2. Cada um dos padrões é processado com base na Ação descrita no registro selecionado para ele no primeiro passo.
3. A lista de padrões traduzidos é compilada em uma nova mensagem, que representa a tradução da mensagem original.
4. Esse método permite que sejam realizadas traduções de todos os tipos, das mais simples até as mais complexas, uma vez que é possível tratar de forma independente cada padrão, podendo inclusive, utilizar um programa, função ou serviço diferente para processar cada um deles.

### **2.6.5. Novas Possibilidades**

Uma vez que Ações diferentes podem ser utilizadas para tratar cada um dos padrões, e essas Ações podem ser algoritmos de qualquer natureza sendo possível gerar inúmeros tipos de saída como documentos multimídia, traduções multi-idiomáticas, gráficos, mapas conceituais, em suma, a possibilidade de poder processar de forma diferente cada um dos padrões cria uma gama enorme de novas possibilidades, tornando o processo de tradução muito mais versátil e poderoso. No capítulo 6 será apresentada uma metodologia para criar essas aplicações e alguns casos de uso serão analisados.

Outro conceito fundamental dessa proposta é a delegação de tarefas. As Ações podem ser aplicadas por qualquer sistema (programas externos, módulos, WebServices etc) que seja capaz de receber uma entrada e retornar uma saída. Isso cria ainda mais possibilidades, permitindo inclusive a utilização de múltiplos sistemas de tradução, para que a melhor técnica possa ser aplicada no tratamento de cada padrão.

Uma pessoa, mesmo sendo leiga em termos de programação de computadores, é capaz de definir um conjunto de condição e ações que representam a forma como ela executaria uma tarefa. Editores amigáveis permitem que ela formalize, facilmente, esse conhecimento em uma especificação, que pode ser interpretada e utilizada para gerar saídas baseadas nesse conhecimento. O método proposto permite que qualquer pessoa crie um sistema computacional capaz de automatizar tarefas, bastando explicitar o seu conhecimento.

### **3. Embasamento Teórico**

A TSL se baseia fundamentalmente em dois conceitos, a delegação de processamento, prática comum nos sistemas que utilizam WebServices (seção 3.9), e nos Sistemas Baseados em Conhecimento (seção 3.8), é uma linguagem declarativa fundada em Regras e Fatos para definir associações do tipo Condição-Ação, de modo a representar o conhecimento necessário para traduzir uma mensagem. Trata-se de uma Linguagem de Marcação (3.2) inspirada no TMX (seção 3.5), formato utilizado para compartilhar Memórias de Tradução.

#### **3.1. Programação Declarativa**

A Programação Declarativa se baseia em descrever o problema e a solução desejada, não a maneira como o programa obterá a solução, um paradigma que desvia o foco do algoritmo, voltando a atenção para a descrição do problema.

Várias linguagens lógicas e funcionais permitem a utilização de técnicas de encapsulamento para suportar uma abordagem declarativa. Porém, uma vez assimilada a idéia, é possível criar em qualquer linguagem de programação, mecanismos capazes de executar especificações declarativas.

As linguagens de marcação são muito úteis nesse tipo de programação, pois fornecem uma maneira simples e eficiente de representar informações e meta-informações. Outros pontos positivos em se utilizar uma linguagem de marcação para representar uma especificação declarativa, é a facilidade de criação e edição, assim como a maneira dinâmica com que novos conceitos podem ser inseridos e representados, bastando definir novas estruturas de marcação (tags).

### 3.2. Linguagens de marcação

*"As Linguagens de Marcação Generalizadas, são fundamentadas no conceito de que a Comunicação é baseada em Símbolos e Regras Estruturadas ou Categorias"* [S.I.F. 2007].

Existem Linguagens de Marcação voltadas para fins específicos, como formatação e descrição de dados, mas existe uma generalização chamada SGML, que é o modelo básico do qual derivam as linguagens de marcação mais utilizadas.

A linguagem de marcação mais popular é a HTML (HyperText Markup Language), uma linguagem de marcação voltada para a formatação de textos e na qual a maior parte das páginas de Internet são escritas atualmente, oferece recursos como links (ligações) e objetos de interação com o usuário,.

### 3.3. SGML

A SGML (Standard Generalized Markup Language), que foi especificada na ISO 8879 em 1986, é uma meta-linguagem que serve de modelo, ou base, para as linguagens de marcação mais utilizadas hoje: HTML, XML etc.

A proposta básica dessa linguagem é fornecer padrões de representação, pelos quais sejam possíveis a geração de documentos que contenham dados, e informações tanto sobre os dados quanto sobre a estrutura do documento.

A legibilidade dos documentos e a facilidade com que podem ser integradas a sistemas computacionais são características das linguagens baseadas na SGML. [Souza-Alvarenga 2004].

### 3.4. XML

O XML (eXtensible Markup Language) é uma versão simplificada do SGML, e é recomendada pela W3C como referência para criação de linguagens de marcação com necessidades específicas, baseada na simplicidade do XML, associado à sua capacidade de representar informações, e meta-informações.

Documentos XML podem armazenar dados e descrição sobre eles, assim como sobre sua estrutura, a qual apesar de ser bem definida e organizada, também permite dinamismo e expansibilidade, o que torna o XML um meio muito prático para trocar dados entre sistemas, são arquivos texto bem legíveis para as pessoas habituadas com a linguagem, seu fácil processamento o torna uma opção muito boa para arquivos de configuração e backup. [Almeida 2002]

### 3.5. O formato TMX

O TMX (Translation Memory Exange) é o padrão aberto de XML para a troca dos dados da memória da tradução. Criado em 1998, TMX é um formato padrão certificado, desenvolvido e mantido por um grupo do LISA(The Localization Industry Standards Association) com interesses especiais, o OSCAR (Open Standards for Container/Content Allowing Re-use). [LISA 2007]

A proposta do TMX é fornecer uma maneira fácil e prática de compartilhar Memórias de Tradução, sem perdas críticas de dados durante o processo. Este formato define uma estrutura e um conjunto de marcações, que formam um padrão capaz de representar: o idioma original da Memória de Tradução, informações sobre o autor, e as entradas relativas aos Fatos que ela é capaz de traduzir.[Simões 2004]

O TMX define alguns conceitos interessantes, como a "tu" (unidade de tradução), que se refere a um agrupamento de padrões equivalentes, em vários idiomas, cada um deles representado como um elemento "seg" (segmento) dentro de uma estrutura "tuv" (variante da unidade de tradução).

A especificação completa e informações adicionais sobre o TMX podem se obtidas a partir do endereço <http://www.lisa.org/tmx>.

Quadro 3.1 - Exemplo de uma entrada TMX

```
<tu srclang="en-US" datatype="html" tuid="sample1">
  <tuv xml:lang="en" datatype="html">
    <seg>link to <bpt i="1" type="link" x="1">
      &amp;a href="www.mysite.com" title="<sub type="x-title">
        my site</sub>"&gt;</bpt>my web site<ept i="1">&lt;/a&gt;,</ept>.
    </seg>
  </tuv>
  <tuv xml:lang="es" datatype="html">
    <seg>enlace a <bpt i="1" type="link" x="1">
      &amp;a href="www.mysite.com/es" title="<sub type="x-title">
        mi sitio</sub>"&gt;</bpt>mi sitio web<ept i="1">&lt;/a&gt;,</ept>.
    </seg>
  </tuv>
</tu>
```

### 3.6. Conhecimento Tácito e Conhecimento Explícito

Pela definição, Conhecimento Explícito é aquele armazenado como informação, na forma de documentos e qualquer outro meio que possa representar o conhecimento. O Conhecimento Tácito é aquele adquirido a partir da interpretação da informação, da utilização do conhecimento adquirido e da socialização.[Nonaka-Takeuchi 1997]

O aprendizado é um processo pessoal, cada indivíduo pode formar um conhecimento distinto a partir da interpretação de uma mesma informação. Isso se deve a diversos fatores, seu embasamento teórico, suas experiências e vários outros fatores subjetivos.

Com base na capacidade humana de adquirir conhecimento tácito a partir de fontes de conhecimento explícito, Nonaka e Takeuchi escreveram sobre a espiral do conhecimento [Nonaka-Takeuchi 1997]. A espiral do conhecimento ilustra e descreve o processo contínuo de aprendizado demonstrando que a evolução do conhecimento depende de uma realimentação, ou seja, conforme o conhecimento é assimilado e desenvolvido é necessário que as evoluções sejam registradas.



Figura 3.1 - Espiral do Conhecimento [Nonaka-Takeuchi 1997]

O conhecimento tácito é pessoal, mas pode ser formalizado e representado em modelo e documentos. Porém, é uma tarefa difícil conseguir representar esse conhecimento, que evolui a cada ciclo da Espiral do Conhecimento.

O primeiro passo para uma captação efetiva do conhecimento individual, é definir um modelo de representação que possa não apenas armazenar informações, mas representar a forma, ou os métodos, utilizados para inferir conhecimento a partir delas.

### **3.7. Bases de Conhecimento**

Uma Base do Conhecimento é equivalente a um banco de dados tradicional, arquivos, registros, informações e seus relacionamentos, porém substituí esses elementos por uma base de Regras e Fatos e por heurísticas, criando uma representação explícita do conhecimento de especialistas no domínio relativo ao sistema [Mendes 1997]. É possível notar claramente a ligação com as Ontologias de Domínio que representam os elementos de domínio e os relacionamentos entre eles, as regras de formação de modelos válidos e outros pontos que também representam o conhecimento de um especialista sobre domínio em questão.

Segundo MENDES, a interação entre o usuário, a base de regras e fatos e a máquina de inferência, que nesse projeto é representada pelo sistema de produção, permite identificar o problema a ser resolvido, as possibilidades de solução e o processo de raciocínio e inferência que levam a conclusões sobre o problema submetido ao sistema.



### 3.8. Sistemas Baseados em Conhecimento

Sistemas Baseados em Conhecimento utilizam métodos que fazem busca em um espaço de possíveis soluções e fazem uso intensivo de heurísticas para tornar a busca efetiva.

São sistemas que realizam inferências baseadas em uma base de conhecimento na qual está representada, de alguma forma, uma ontologia que descreve o domínio do problema a ser resolvido. Esses sistemas raciocinam sobre suas possíveis ações no mundo e que conhecem o estado atual do mundo, como este evolui, sabe como identificar estados desejáveis do mundo e avaliar o resultado das ações [Canuto 2000].



Figura 3.2 - Como o ser humano identifica o domínio de um problema [Canuto 2000].

Segundo CANUTO, o ser humano resolve problemas aplicando seus conhecimentos a ele, escolhendo o mais interessante dentre os modelos válidos que pode inferir com base do que conhece sobre o domínio do problema e os Sistemas Baseados em Conhecimento imitam esse padrão de comportamento para resolver os problemas.

A Base de Conhecimento contém as regras e fatos que se aplicam ao domínio do problema e as representações de ações e acontecimentos do mundo enquanto a Máquina de Inferência é responsável por utilizar essa base para gerar os modelos válidos que representam as possíveis soluções do problema.

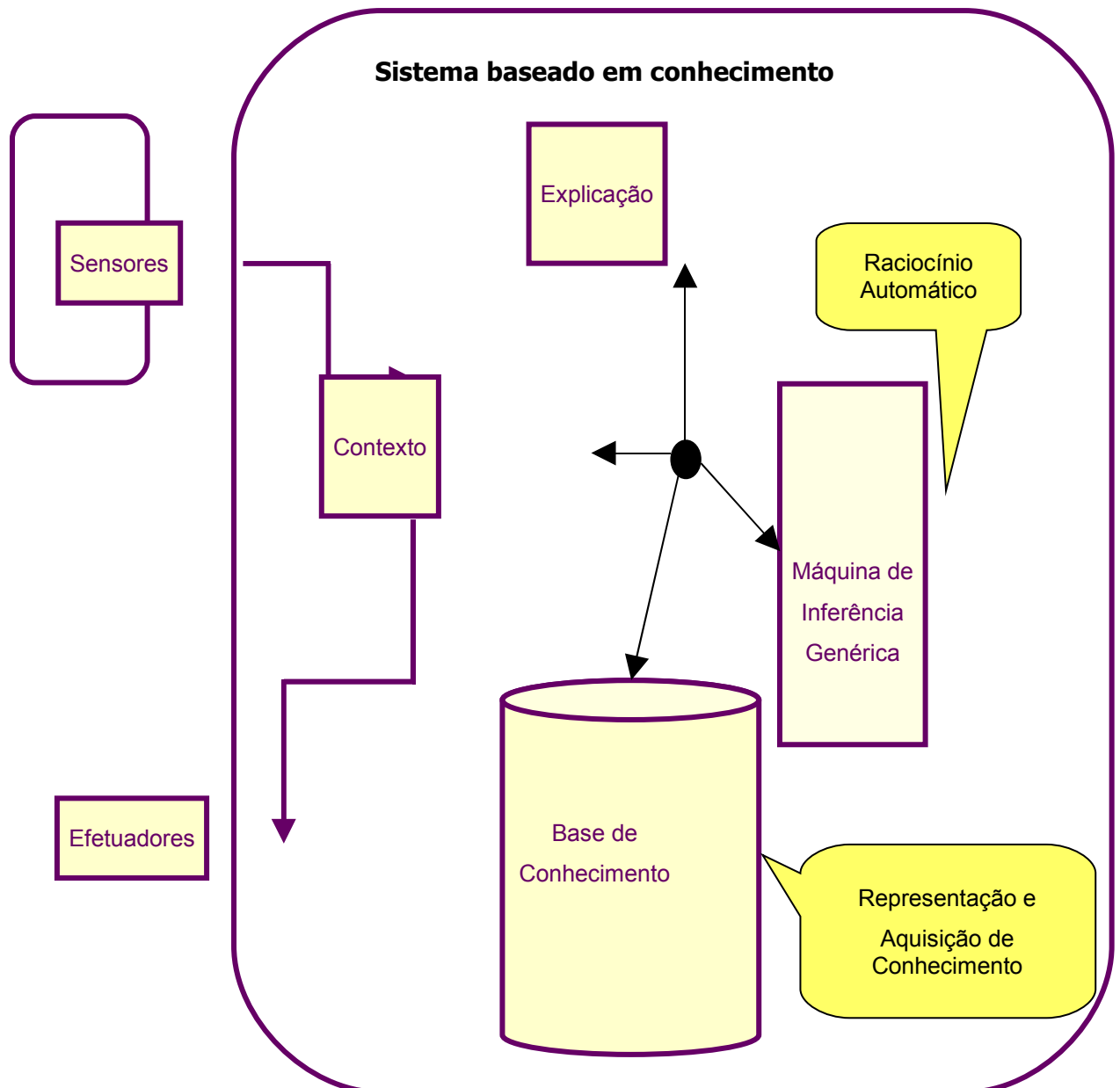


Figura 3.3 -Estrutura de um sistema baseado em conhecimento [Canuto 2000]

### **3.9. WebServices**

Segundo Menéndez [Menéndez 2002], WebService é uma aplicação que aceita solicitações de outros sistemas através da Internet, é possível acessar uma funcionalidade de uma aplicação que está sendo executada em um computador remoto, sem a necessidade de criar uma instância local.

A arquitetura proposta para a implementação dos WebServices se baseia no XML no que diz respeito à troca de mensagens, o que torna simples que se descreva de forma declarativa a lógica envolvida no processo [Martin 2005].

O que torna os WebServices tão atrativos, é a possibilidade de se delegar processamento. Aplicações desenvolvidas na forma de especificações declarativas podem determinar que diferentes serviços Web sejam utilizados no decorrer da execução do programa.

### 3.10. PHP

"PHP é uma linguagem script onde sua execução é feita no lado servidor (Server-Side Script), é Open Source e multiplataforma" [Gomes 2006].

PHP é uma sigla recursiva (PHP: Hypertext Preprocessor), assim como GNU (GNU's Not Unix). Trata-se de uma brincadeira comum entre os membros das comunidades relacionadas à filosofia de software livre. É uma linguagem que suporta os paradigmas Procedural e Orientado a Objetos, e embora possua uma sintaxe muito simples, fornece poderosos recursos, sobretudo para composição de aplicações WEB.

Essa linguagem permite uma criação bem rápida de protótipos, permitindo que sejam usadas páginas WEB como interface com o usuário e exibindo em tempo real as alterações realizadas no código fonte do sistema. A saída padrão dos sistemas escritos em PHP são páginas HTML, tornando trivial a inclusão de objetos multimídia e formatações dos textos.

Os recursos oferecidos para manipular textos e trocar mensagens são outros pontos, e o enorme número de usuários, que compartilham informações e módulos escritos nessa linguagem, fazem com que seja uma boa escolha para a programação de sistemas que precisam ser concluídos em um curto espaço de tempo.

Um dos pontos negativos do PHP é o desempenho, pois é uma linguagem interpretada por um módulo instalado em um servidor WEB. Porém, como a prioridade no desenvolvimento do protótipo é avaliar o resultado, isso não foi um fator que desestimulasse a opção.

**Estão disponíveis gratuitamente no sítio do PHP, localizado no endereço [www.php.net](http://www.php.net), o manual completo da linguagem, exemplos, fóruns de discussão e todas as ferramentas necessárias para a construção e execução de sistemas PHP.**

## **4. A TSL**

A TSL (Translator Specification Language) é uma linguagem declarativa que permite a um especialista de determinada área, representar seu conhecimento de uma forma que pode ser utilizado para realizar processamentos automáticos.

### **4.1. O Conceito da Linguagem**

O conceito básico da TSL, é que o computador pode utilizar o conhecimento explícito para ações sobre um texto de entrada e conforme a base de conhecimento cresce, é possível gerar traduções cada vez mais semelhantes às geradas pelos tradutores humanos que alimentam a base.

Em uma especificação TSL, não se define como executar as ações sobre um padrão, mas sim como reconhecê-los e quais ações aplicar sobre eles. A aplicação das ações fica por conta de sistemas externos, utilizados pelo interpretador para processar os padrões identificados.

A delegação de tarefas (ações/processamento), permitindo que programas externos processem os padrões reconhecidos, de forma independente, torna possível utilizar o método de tradução apresentado na seção 2.6 para criar diversos tipos de sistemas computacionais, sendo que cada um desses sistemas pode ser descrito na forma de uma especificação TSL.

## 4.2. A Gramática

Serão apresentados nesse capítulo os itens léxicos que são válidos na TSL. Sempre que for relevante, será apresentada uma breve descrição da semântica associada ao item léxico em questão.

**<NUM>** ::= [0-9]

Esse item se refere aos caracteres numéricos do teclado, ou seja, do 0 ao 9.

**<LETRA>** ::= [A-Z a-z]

Podem ser letras maiúsculas ou minúsculas. Não admite caracteres especiais.

**<CARACTER>** ::= [<NUM>,<LETRA>]

Essa generalização define uma lista de símbolos válidos para formar itens STRING.

**<STRING>** ::= [<LETRA>]+ [<CARACTER>]\*

String é uma cadeia não vazia de itens CHARACTER, iniciada por uma LETRA.

**<TEXTO>** ::= [STRING]\*

Um texto pode armazenar qualquer conjunto de caracteres, incluindo um conjunto vazio. Esse item é utilizado para representar descrições, padrões e outras formações compostas por seqüências de palavras.

**<CLASSE>** ::= ["ART", "SUB", "ADJ", "ADV", "PRE", "VER" etc]

Esse item se refere às abreviaturas das classes gramaticais. As abreviaturas admitidas como valor de itens do tipo CLASSE são definidas nas gramáticas disponíveis para o interpretador.

**<GRASEQ>** ::= [<CLASSE>]\*

O item GRASEQ é utilizado para representar padrões formados por mais de um item CLASSE. Itens GRASEQ são utilizados para representar padrões que se referem a regras gramaticais.

**<USUARIO>** ::= <STRING>

A informação contida nesse item se refere a um nome de usuário, logo, não pode ser vazio, e necessariamente deve começar com uma LETRA.

**<HOST>** ::= [<LETRA>](2-3)"."[<LETRA>](2-3)

HOST se refere ao domínio de rede, do tipo “.com.br” ou “.co.jp” .

**<EMAIL>** ::= <USUARIO>“@”<HOST>

Este item, no cabeçalho de uma especificação TSL, tem seu significado associado ao e-mail de contato do autor.

**<DIA>** ::= [<NUM>](2)

Dia do mês (1 a 31) em que foi criada/atualizada a especificação TSL.

**<MES>** ::= [<NUM>](2)

Mês do ano, admite valores de 1 a 12, representando os meses do ano.

**<ANO>** ::= [<NUM>](4)

O ano deve ser registrado com 4 dígitos. Ex: 1992, 2007.

**<DOMINIO> ::= <STRING>**

O item DOMINIO no CABECALHO da especificação tem semântica associada ao nome do domínio no qual devem estar representados os textos fornecidos como entrada, enquanto em um item TA, referência o domínio no qual o átomo de tradução representa o PADRAO reconhecido.

**<DESCRICAO> ::= <TEXTO>**

O significado do item DESCRICAO é uma curta explicação sobre o DOMINIO relacionado e o resultado gerado pela especificação TSL.

**<NOME> ::= <USUARIO> <TEXTO>**

A semântica associada a esse item é o nome do criador da especificação TSL. Pode ser um nome de usuário (login, apelido) ou o nome completo do autor.

**<PADRAO> ::= <TEXTO>**

A informação contida nesse item tem como significado o novo segmento de texto que deve ser utilizado no lugar de um padrão identificado. Esse item também pode ser utilizado para aplicar ações simples, como inserção de novas marcações.

Também pode ser utilizado para organizar os parâmetros e formatar as entradas nos casos em que é necessário utilizar aplicações externas para aplicar ações que gerem o resultado desejado.



**<SERVICO>** ::= <TEXTO>

Caso seja necessária a aplicação de ações ao PADRAO, o item SERVICO informa o nome pelo qual a aplicação responsável por aplicá-las é acessível para o sistema. Um valor vazio para o item SERVICO indica que nenhum processamento é necessário sobre o padrão identificado.

**<ORIGEM>** ::= <ORIGIN> <TEXTO> </ORIGIN>

Este campo representa um padrão a ser procurado na entrada. Nos fatos não existe restrição quanto à formação do texto contido em <ORIGEM>, nas regras, porém, esse texto deve ser formado por concatenações de classes gramaticais como: ADJ,ART,SUB,ADV ... O significado disso é que nas regras se procura por padrões explícitos no texto, enquanto nas regras a busca é feita analisando a classe gramatical das palavras em busca de padrões do tipo <GRASEQ>.

**<ATOMO>** ::= <TA>

<DOM> <DOMINIO> </DOM>

<PATTERN> <PADRAO> </PATTERN>

<SERVICE> <SERVICO>“</SERVICE>

</TA>

O ATOMO é uma unidade que carrega a informação da possibilidade de traduzir o padrão reconhecido pelo caso onde está contido, para o domínio descrito em DOMINIO, utilizando para isso a aplicação do sistema representado em SERVICO, sobre o texto representado em PADRAO.

```

<CABECALHO> ::= <HEAD>
                <AUTHOR> <NOME> </AUTHOR>
                <CONTACT> <EMAIL> </CONTACT>
                <DATE> <DIA> "-" <MES> "-" <ANO> </DATE>
                <DOM> <DOMINIO> </DOM>
                <DESC> <DESCRICAO> </DESC>
            </HEAD>

```

Tem como objetivo identificar o autor, a data de criação/atualização do arquivo e o domínio no qual essa especificação TSL aceita os textos de entrada.

```

<CASO> ::= <CASE> <ORIGEM> [<ATOMO>]+ </CASE>

```

O CASO é um item que se refere a uma estrutura que é formada por um padrão a ser reconhecido (ORIGEM) e por uma lista de TA`s, cada um deles trazendo a possibilidade de traduzir ORIGEM para o DOMINIO representado nele.

```

<CORPO> ::= [<CASO>]*

```

O CORPO é o item que trás informações relativas ao conhecimento contido na especificação TSL. Ele possui uma lista de todos os itens do tipo CASO.

```

<ESPECIFICACAO> ::= "<TSL> <CABECALHO> <CORPO> </TSL>"

```

O item ESPECIFICACAO representa todo o conjunto de itens que forma uma especificação TSL. O cabeçalho (head) com as informações sobre a especificação e o autor e o corpo (body) que contem o conhecimento armazenado acessível à especificação.

### **4.3. A Estrutura**

A estrutura de uma especificação TSL é muito simples, o que permite uma boa legibilidade, assim como uma facilidade de criação e edição. É composta por 2 seções, a Head e a Body.

Todos os campos contidos no TSL foram descritos na seção 4.2, logo, as descrições sobre a estrutura serão breves, e haverá um exemplo na seção 4.4 que ilustrará toda a estrutura de uma especificação TSL.

#### **4.3.1 Seção Head**

A seção Head é o cabeçalho, diz respeito ao cabeçalho da especificação TSL. Nela estão armazenados os dados que identificam o autor da especificação, o domínio no qual devem estar os textos de entrada, e uma descrição que pode ser utilizada para explicar a finalidade da especificação, ou seja, o resultado que ela produz quando aplicada sobre um texto.

#### **4.3.2. Seção Body**

A seção Body é o corpo da especificação TSL, ela é formada por uma lista de casos, cada um delimitado pelas marcações <CASE> e </CASE>.

Essa seção é onde está armazenado todo o conhecimento que foi explicitado na especificação TSL.

#### **4.3.3. Registro de Caso**

Cada Caso (seção 2.5.2.) é responsável pelo reconhecimento de um determinado padrão (Fato), ou conjunto de padrões (Regras), e por determinar a forma como esse padrão deve ser tratado. Para isso, cada Caso conta com uma lista de Átomos de Tradução.

#### 4.3.4. Átomo de tradução

Um Átomo é delimitado pelas marcações <TA> e </TA>. Cada um dos Átomos de Tradução de um caso é responsável por determinar como o padrão reconhecido pelo caso será representado. O atributo <DOM> determina o domínio, ou seja, o contexto no qual o átomo será selecionado.

Quando é selecionado um que Caso reconhece um padrão, o Átomo de Tradução selecionado é aquele que tem o atributo <DOM> referente ao domínio escolhido para a representação destino da tradução.

#### 4.4. Um exemplo de especificação TSL

Nessa seção será apresentado um exemplo de especificação TSL que executa traduções do Português para alguns outros idiomas. Nesse exemplo, ações diferentes são realizadas em pelos TA`s de um mesmo caso. Por conta disso, ao se traduzir para idiomas diferentes, à formatação da saída também fica diferente. Essa é apenas uma ilustração da versatilidade da TSL.

Como é um exemplo bem simples, apenas para ilustrar a estrutura do TSL, é obvio que ela contempla apenas alguns poucos casos, não permitindo uma tradução completa.

Quadro 4.1 - Exemplo de uma Especificação TSL

```

<TSL>
  <HEAD>
    <AUTHOR>Marcello Novaes de Amorim</AUTHOR>
    <CONTACT>novaes@inf.ufes.br</CONTACT>
    <DATE>14-02-2007</DATE>
    <DOM>PORTUGUÊS-BR</DOM>
    <DESC>Tradutor de Idioma português para alguns outros idiomas</DESC>
  </HEAD>
  <BODY>
    <CASE>
      <PATTERN>PEGA LEVE</PATTERN>
      <TA>
        <DOM>PORTUGUÊS FORMAL</DOM>
        <PATTERN>Tenha calma</PATTERN>
        <SERVICE>NEGRITO</SERVICE>
      </TA>
      <TA>
        <DOM>INGLÊS COLOQUIAL</DOM>
        <PATTERN>Take it easy</PATTERN>
        <SERVICE>NULL</SERVICE>
      </TA>
      <TA>
        <DOM>ESPAÑHOL COLOQUIAL</DOM>
        <PATTERN>Coge ha conducido</PATTERN>
        <SERVICE>DESTAQUE</SERVICE>
      </TA>
    </CASE>
  </BODY>
</TSL>

```

Quadro 4.1 (continuação) - Exemplo de uma Especificação TSL

```

<CASE>
  <PATTERN>[SUBSTANTIVO] [ADJETIVO]</PATTERN>
  <TA>
    <DOM>INGLÊS</DOM>
    <PATTERN>[ADJETIVO] [SUBSTANTIVO]</PATTERN>
    <SERVICE>NULL</SERVICE>

<CASE>
  <PATTERN>ZÉ NINGUÉM</PATTERN>
  <TA>
    <DOM>INGLÊS COLOQUIAL</DOM>
    <PATTERN>John Doe</PATTERN>
    <SERVICE>DESTAQUE</SERVICE>

  </TA>
  <TA>
    <DOM>PORTUGUÊS COLOQUIAL</DOM>
    <PATTERN>Indigente</PATTERN>
    <SERVICE>NEGRITO</SERVICE>

  </TA>
  <TA>
    <DOM>INGLÊS FORMAL</DOM>
    <PATTERN>NULL</PATTERN>
    <SERVICE>NEGRITO</SERVICE>

  </TA>
  <TA>
    <DOM>ESPANHOL</DOM>
    <PATTERN>NULL</PATTERN>
    <SERVICE>Dicionario_Portugues_Espanhol</SERVICE>

  </TA>
  <TA>
    <DOM>ITALIANO</DOM>
    <PATTERN>NULL</PATTERN>
    <SERVICE>Dicionario_Portugues_Italiano</SERVICE>

  </TA>
</CASE>
</BODY>
</TSL>

```

## 5. Interpretadores TSL

Um interpretador TSL é um ambiente virtual no qual é possível aplicar o conhecimento descrito em especificações TSL para gerar traduções de textos.

### 5.1. Requisitos

- Reconhecer a estrutura da TSL
- Ter acesso a gramáticas relativas aos domínios das especificações.
- Ser capaz de identificar os Casos contidos em uma especificação TSL e utilizá-los para identificar padrões em um texto.
- Ser capaz de definir uma ordem de prioridade, para que sempre seja selecionado o Caso mais adequado.
- Ser sensível ao contexto podendo identificar qual dos Átomos de Tradução do Caso selecionado deve ser utilizado de acordo com o domínio destino.
- Poder acessar outras aplicações(WebServices etc) e utilizá-las para processar adequadamente cada um dos padrões identificados.
- Ser capaz de reunir de forma correta as saídas parciais do processamento de cada padrão, produzindo um único documento de saída.

## **5.2. Caso Padrão**

No caso ideal, ou seja, quando o conhecimento explícito contido na especificação TSL é suficiente para agrupar todo o texto em padrões reconhecidos, o método descrito na seção 2.6 é suficiente para gerar uma tradução completa. Porém, é necessário utilizar algum recurso para tratar os fragmentos do texto de entrada que não são identificados como um padrão associado a um Caso.

Uma solução simples é adotar um Caso Padrão, ou seja, os fragmentos que não são reconhecidos, são associados a um Caso Padrão que tem a mesma estrutura de um Caso comum, com informações sobre como os padrões associados a ele devem ser processados.



### 5.3. O protótipo

Nessa seção é descrito o processo de desenvolvimento do protótipo do interpretador utilizado durante a concepção da TSL. São discutidos alguns aspectos específicos desse sistema, porém, o objetivo central é mostrar como a criação de um protótipo mais simples acelerou os testes, permitindo, em pouco tempo, uma formalização de uma versão oficial da TSL, e como o produto final revelou os requisitos necessários para um interpretador TSL, independente da implementação.

A criação de um protótipo em PHP se mostrou uma boa alternativa pela alta produtividade, facilidade em manipular textos e gerar saídas multimídia voltadas para WEB. Por se tratar de uma linguagem interpretada, os resultados das alterações podem ser observados em tempo real, permitindo a avaliação imediata de cada alteração da TSL. A facilidade na manipulação de textos e do acesso a sistemas externos permitiu que praticamente todo o esforço fosse dedicado ao desenvolvimento da linguagem, pois a implementação do protótipo foi trivial, em sua maior parte.

A interface do protótipo é baseada em páginas WEB o que agilizou ainda mais o desenvolvimento do protótipo. Essa característica também possibilitou que elementos multimídia como imagens e vídeos fossem facilmente incorporados às saídas geradas, uma vez que os navegadores de Internet já suportam a exibição desse tipo de arquivo.

A página inicial, representada na figura 5.1, permite ao usuário selecionar o Caso Padrão, o Domínio Destino, e a Especificação TSL que deseja utilizar para gerar a tradução. Isso configura o interpretador, fazendo com que ele se comporte como uma aplicação WEB, que processa entradas de acordo com a especificação selecionada. O usuário pode escolher se deseja que sejam exibidas informações adicionais sobre o processo de tradução, ou apenas o resultado final.

## Interpretador TSL - Protótipo 1.0

Selecione a aplicação TSL

Tradutor (Origem PT-BR) ▼

Selecione o serviço padrão

Dicionario\_Portugues\_Ingles ▼

Selecione o domínio destino

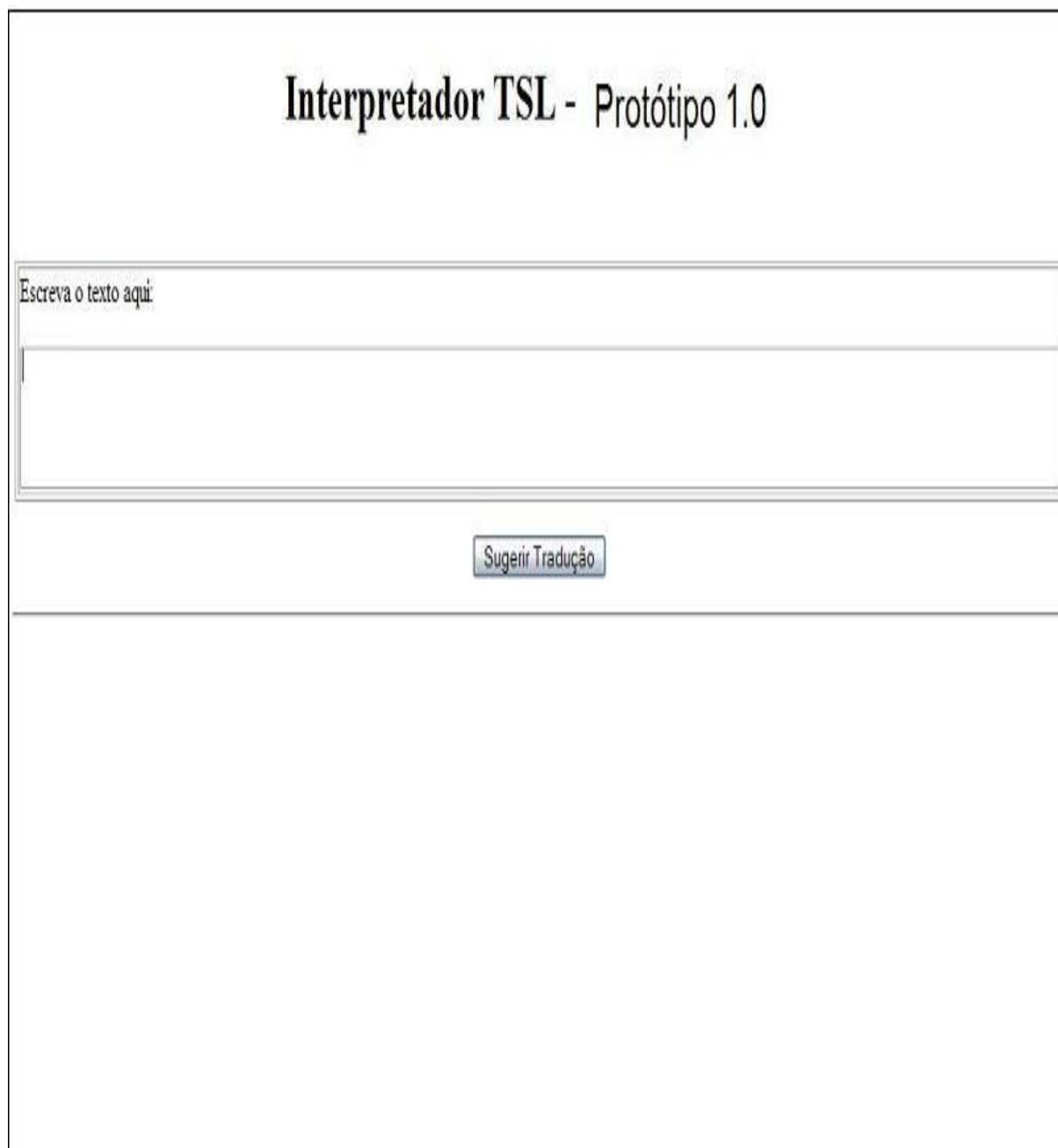
INGLÊS COLOQUIAL ▼

Executar

Casos do TSL |  Saída |  Casos Reconhecidos

Figura 5.1 – Tela inicial do protótipo

Em seguida, o sistema leva o usuário a uma outra página (figura 5.2), na qual ele pode utilizar uma janela de texto, para gerar traduções de acordo com a especificação escolhida.



Interpretador TSL - Protótipo 1.0

Escreva o texto aqui:

Sugerir Tradução

Figura 5.2 – Tela de uso do sistema

### 5.3.1. Obtendo os dados de uma especificação TSL

O primeiro passo para que um interpretador TSL seja capaz de gerar uma tradução a partir de um texto de entrada, é o reconhecimento das informações contidas na especificação TSL escolhida. O algoritmo utilizado pode ser generalizado pela seqüência de passos:

- Ler todo o conteúdo da especificação TSL;
- Separar em duas partes, uma contendo o cabeçalho e outra o corpo;
- Popular uma estrutura de dados com as informações de cabeçalho;
- Identificar todos os Casos contidos no corpo da especificação;
- Identificar todos os átomos de tradução de cada um dos Casos;
- Criar uma lista de estruturas de dados, cada uma delas contendo as informações sobre o padrão reconhecido pelo caso uma lista com os átomos de tradução associados ao Caso.
- A lista de átomos de tradução contida em cada estrutura que armazena um Caso, deve conter em cada uma de suas posições uma estrutura de dados que armazene o domínio para no qual o TA pode representar o padrão reconhecido, o padrão que deve ser utilizado em substituição ao encontrado, e qual o sistema responsável por aplicar as ações semânticas necessárias.

Como a TSL é uma linguagem de marcação, foi possível utilizar um recurso do PHP chamado “explode”, que utiliza um determinado padrão separador para segmentar um texto em uma lista de sub-textos. Aplicando esse recurso de forma sucessiva, foi criado um “parser” que é utilizado para extrair todas as informações de uma especificação TSL, armazenando todas elas em estruturas de dados de fácil acesso.

Na figura 5.3 podem ser visualizadas as informações extraídas pelo protótipo de uma especificação TSL. Pode-se notar na parte superior da figura os dados do cabeçalho, e na página são exibidos os Casos, delimitados pelas linhas diferenciadas, sendo que cada seqüência de linhas brancas representa a lista de TA's de cada Caso.



Caso	DE [NUMERO] ATÉ [NUMERO]	
Dominio	Ação	Serviço
INGLÊS COLOQUIAL	NULL	CONTADOR
Caso	PEGA LEVE	
Dominio	Ação	Serviço
PORTUGUÊS FORMAL	Tenha calma	NEGRITO
INGLÊS COLOQUIAL	Take it easy	NULL
ESPAÑHOL COLOQUIAL	Coge ha conducido	DESTAQUE
Caso	ISSO É UM TESTE	
Dominio	Ação	Serviço
INGLÊS FORMAL	NULL	Dicionario_Portugues_Ingles
Caso	[SUBSTANTIVO] [ADJETIVO]	
Dominio	Ação	Serviço
INGLÊS FORMAL	[ADJETIVO] [SUBSTANTIVO]	Dicionario_Portugues_Ingles
INGLÊS COLOQUIAL	[ADJETIVO] [SUBSTANTIVO]	DESTAQUE
PORTUGUÊS REBUSCADO	[ADJETIVO] [SUBSTANTIVO]	NULL
Caso	ZÉ NINGUÉM	
Dominio	Ação	Serviço
INGLÊS COLOQUIAL	John Doe	DESTAQUE
PORTUGUÊS COLOQUIAL	Índigente	NEGRITO
INGLÊS FORMAL	NULL	NEGRITO
ESPAÑHOL	NULL	Dicionario_Portugues_Espanhol
ITALIANO	NULL	Dicionario_Portugues_Italiano

Figura 5.3 – Informações obtidas de uma especificação TSL

### **5.3.2. Acessando a gramática correta**

Uma das informações contidas no cabeçalho é o domínio no qual estarão representados os textos de entrada. A partir desse dado, o sistema seleciona a gramática que deve utilizar para determinar a classe gramatical de cada uma das palavras do texto.

O protótipo faz isso de uma maneira muito simples. Existe no sistema uma pasta contendo uma série de arquivos, cada um deles armazenando dados relativos a uma gramática e pode ser editado conforme se deseja adicionar novas entradas, alterar ou remover algumas das existentes. De acordo com a informação obtida no cabeçalho da especificação TSL, o arquivo correto é selecionado pelo nome.

### **5.3.3. Tradução sensível ao contexto**

Na figura 5.1 é possível observar a opção “Selecione o domínio destino”. A informação obtida por meio desse campo do formulário é utilizada para determinar o domínio no qual será gerada a tradução.

O protótipo utiliza esse dado para selecionar o átomo de tradução adequado em cada Caso selecionado durante o processo. Isso é feito de maneira muito fácil, bastando comparar o conteúdo desse campo com o valor contido no atributo DOM dos TA's encontrados na lista de átomos. Quando não existe no Caso nenhum átomo de tradução com o atributo DOM idêntico ao domínio destino selecionado, o padrão é deixado inalterado, e é encaminhado para o serviço padrão selecionado no formulário inicial.

Observando a figura 5.3 pode-se visualizar o valor do atributo DOM dos TA's de cada um dos Casos na coluna identificada pelo rótulo “Domínio”.

#### **5.3.4. Acesso a aplicações externas**

Na seção 5.3.1 foi explicada a cena que pode ser observada na figura 5.3. A página exibe uma lista de Casos, delimitados por linhas amarelas onde aparecem os padrões reconhecidos por cada um, seguido de uma seqüência de linhas brancas, cada uma representando um dor átomos de tradução da lista contida no Caso.

Existem três colunas na tabela de TA's, identificadas pelos rótulos Domínio, Ação e Serviço.

A informação na coluna Ação informa o padrão que deve ser utilizado no lugar do identificado na mensagem original por meio do Caso. Essa parte pode ser vista como a Memória de Tradução do sistema. Quando aparece o valor NULL na coluna Ação, significa que o padrão original deve ser mantido.

A coluna Serviço informa qual aplicação externa deve ser utilizada para aplicar as ações para que o padrão seja traduzido. É importante ressaltar que essas ações externas são aplicadas sobre o padrão encontrado na coluna Ação, dessa forma, quando aparece o valor NULL na coluna Serviço, é utilizado o "Serviço Padrão", selecionado na página inicial.

### 5.3.5. O processo de tradução

A implementação em PHP permitiu que o foco fosse voltado para o desenvolvimento da linguagem por possuir características que facilitam muitos aspectos de implementação, o método que identifica os padrões no texto demonstra isso.

Existe no PHP um recurso que pode ser utilizado para converter um texto em um vetor de palavras, é utilizado para gerar uma representação mais prática dos padrões contidos nos Casos. Essa abordagem permitiu a implementação de um método de busca, que embora não tenha um bom desempenho, é eficaz e dispensa a necessidade de mecanismos de “backtracking” durante o processo.

Um ponto importante a ser observado antes de iniciar a explicação do processo de busca, é que o padrão armazenado nos Casos pode ser composto por palavras e “classes gramaticais” (Classes), que podem ser identificadas por aparecerem entre colchetes.

A gramática selecionada segundo a explicação apresentada na seção 5.3.2 fornece uma lista das palavras válidas no domínio base, ou seja, o especificado no cabeçalho, e as suas classes gramaticais. Como se trata de um protótipo, não foi levado em consideração nenhum caso em que uma palavra pudesse assumir mais de uma classe gramatical. Durante as comparações, os Casos recebem uma pontuação que aumenta de acordo com o número de palavras e diminui para cada Classe contida no padrão, dessa forma, os Fatos são priorizados. A “penalidade” aplicada para cada Classe contida no padrão pode ser configurada no código, e existe a possibilidade de incluir essa configuração no nas próximas versões da linguagem.

O método de busca conta com dois apontadores, um deles indica o início do segmento de texto que ainda não foi traduzido e o outro aponta para o Caso que alcançou uma maior pontuação a partir desse ponto.



Inicialmente o ponteiro referente à posição inicial é apontado para a primeira palavra do texto, e até que se alcance o final do texto é realizada a seqüência de passos a seguir:

1. Percorre-se a lista de Casos, separando aqueles que se inicia com a palavra inicial.
2. Dos Casos selecionados verifica-se quais deles podem ser “encaixados” no texto a partir da posição inicial.
3. Se nenhum Caso for adequado, a palavra inicial é acumulada numa lista e a posição inicial é incrementada, passando para a próxima palavra e volta-se para o passo 1, a menos que se chegue ao final do texto, nesse caso, seleciona-se o “Caso Padrão” como o Caso escolhido e em seguida pula-se para o passo 6.
4. Se um ou mais Casos forem “encaixados” com sucesso, o de maior pontuação é selecionado. Atualiza-se a posição inicial para a próxima palavra após o padrão encontrado.
5. Processa-se a lista acumulada com as palavras que não se encaixaram em nenhum caso utilizando o “Caso Padrão” e adiciona-se o resultado à saída.
6. Processa-se o padrão encontrado de acordo com o Caso escolhido e em seguida adiciona-se o resultado à saída.
7. Avalia-se a posição inicial, caso o texto tenha terminado, encerra-se o laço, e exhibe-se a saída. Se ainda existem palavras a serem traduzidas, volta-se ao passo 1.

A saída é exibida na forma de uma página de Internet, como um documento multimídia que representa uma tradução da mensagem original.

## **6. Programas TSL**

A TSL é uma linguagem que fornece meios para a especificação de tradutores baseados no método apresentado na seção 2.6, que permite a identificação de padrões em um texto e a aplicação de um conjunto distinto de ações semânticas a cada um deles.

Essa forma de executar traduções torna possível a criação de diversos tipos de representações e permitir que sistemas externos processem partes do texto de entrada, viabilizando a especificação de tradutores capazes de executar virtualmente qualquer tipo de tratamento sobre a mensagem original.

Como uma especificação TSL pode representar um tradutor capaz de realizar diversos tipos de processamento, pode-se dizer que um interpretador TSL se torna um diferente programa durante a execução de cada especificação.

Em suma, um Programa TSL é um sistema capaz de processar entradas de texto seguindo uma especificação.

### **6.1. Os Conceitos**

Um programa TSL se baseia em dois conceitos básicos, o processamento de padrões de forma individual e independente, e a delegação de processamento para outros sistemas computacionais.

O processamento individual e independente permite que os padrões identificados sejam tratados como parâmetros de uma aplicação e a delegação de tarefas permite a utilização de sistemas existentes, cada um deles nas situações em que fornecem melhor resultado. Pode-se, por exemplo, criar uma programa TSL que executa um tradução do português para o inglês, enviando cada padrão para ser processado por um sistema que apresenta melhor resultado naquele caso específico.

## **6.2. Como programar em TSL**

O processo de criação de especificações TSL é simples, basta definir a lista dos Casos capazes de identificar todos os padrões que podem aparecer em um texto e determinar as ações que devem ser aplicadas em cada caso. A dificuldade está em se determinar quais são esses casos, e em encontrar uma forma de explicitar esse conhecimento.

Uma metodologia baseada na espiral do conhecimento, apresentada na seção 3.6, pode ser aplicada para facilitar a especificação desses programas em TSL. Ela é composta pela seqüência de passos:

1. Definição do escopo;
2. Elicitação do conhecimento do especialista;
3. Determinar as ações do Caso Padrão;
4. Formalização dos Casos;

Esses passos serão detalhados nos sub-tópicos dessa seção.

### **6.2.1. Definição do escopo**

O primeiro passo é determinar o que o programa TSL pode receber como entrada, e quais as possíveis saídas que ele deve ser capaz de fornecer. Determinar os domínios em que devem estar representadas as mensagens originais e para quais domínios elas poderão ser traduzidas.

Esse passo delimita as competências da aplicação e define as restrições das entradas fornecidas.

### **6.2.2. Elicitação do conhecimento do especialista**

O método utilizado pelos interpretadores TSL visa simular o comportamento de um especialista no domínio associado à tarefa que será executada. Para tanto é necessário que o sistema saiba o que uma pessoa capacitada faria em cada caso que pode ocorrer durante o processo de tradução.

Uma forma interessante de coletar as informações necessárias é observar um especialista executando a tarefa pretendida, anotando quais as ações executadas por ele em cada situação.

### **6.2.3. Determinar as ações do Caso Padrão**

Obviamente, é muito difícil que em um primeiro momento consiga-se identificar todos os Casos necessários pra que a tradução seja semelhante à gerada manualmente. Dessa forma, é normal que existam partes do texto que não sejam reconhecidos por nenhum dos Casos.

Uma estratégia para melhorar a qualidade do resultado final é definir qual o conjunto de ações mais utilizado, ou um conjunto genérico de ações, e aplicá-lo aos fragmentos de texto que não forem reconhecidos pelos Casos. Esse conjunto de ações compõe o Caso Padrão.

### **6.2.4. Formalização dos Casos**

Uma vez identificadas às ações que devem ser aplicadas em cada situação deve-se formalizar esse conhecimento. A TSL fornece uma maneira simples de representar cada uma dessas situações por meio de registros do tipo Caso.

De uma forma pratica, a formalização de um Caso consiste em descrever o padrão que ao ser encontrado no texto, deve ser associado ao registro, e em cada átomo de tradução do mesmo são descritas as ações que devem ser aplicadas para realizar uma tradução do padrão identificado para um determinado domínio.

### 6.3. Estudo de Caso

Os estudos de caso têm como objetivo demonstrar as características e a validade da TSL. Todos os exemplos foram testados no protótipo apresentado na seção 5.3, logo, os parâmetros que podem ser fornecidos pelo usuário, além do texto de entrada e da especificação TSL a ser usada, são:

- Domínio da Entrada;
- Domínio Destino;
- Serviço Padrão;

Os serviços disponíveis para cada um dos exemplos serão diferentes, de acordo com o contexto do problema.

### 6.3.1. Tradutor Multi-Idiomático

#### Serviços disponíveis:

- **Dicionário de palavras:** um serviço que utiliza o idioma origem e destino selecionados para selecionar um dicionário de palavras. Recebe como entrada uma lista de palavras, e retorna uma lista de palavras traduzidas, respeitando a ordem.
- **Dicionário do Google:** o google disponibiliza um dicionário de sugestões, que embora não seja adequado para traduções formais, funciona bem boa parte das vezes.
- **Negrito:** Recebe como entrada uma seqüência de palavras e retorna a mesma seqüência, porém, destacadas em negrito.

**Escopo:** Um sistema que aceita textos em diversos idiomas, e fornece como saídas traduções desses textos em um idioma-destino, fornecidos na página inicial do sistema (Figura 5.1).

**Elicitação do conhecimento do especialista:** Este é um típico caso de sistema genérico, ou seja, serve para um universo muito grande de casos. A dificuldade em fazer o levantamento dos Casos para esse sistema vem da necessidade de encontrar pessoas que conheçam vários pares de idiomas ou de diversas pessoas que conheçam ao menos um par de idiomas. Para ilustrar a elicitación do conhecimento nesse caso, será considerado que existe um especialista em tradução do português para o inglês e o espanhol.

Existem dois grupos de padrões a serem identificados, as regras gramaticais (Regras), e as exceções (Fatos) como expressões idiomáticas, gírias etc. Facilmente se observa o grande numero de Casos necessários para que seja possível gerar uma tradução semelhante à manual, logo, esse é um sistema que melhora sensivelmente conforme são adicionados novos Casos na especificação.

**Determinar as ações do Caso Padrão:** No Caso Padrão será utilizado o tradutor do Google, pois mesmo sendo um tradutor de sugestões, ele fornece um resultado melhor do que o obtido traduzindo palavra por palavra.

**Formalização dos Casos:** O quadro 6.1 apresenta dois Casos, um deles representando uma Regra, e um outro com um Fato. O domínio de entrada será o “Português Coloquial”.

Quadro 6.1 - Exemplo de Casos TSL

```

<CASE>
  <PATTERN>PEGA LEVE</PATTERN>
  <TA>
    <DOM>INGLÊS COLOQUIAL</DOM>
    <PATTERN>TAKE IT EASY</PATTERN>
    <SERVICE>NEGRITO</SERVICE>
  </TA>
  <TA>
    <DOM>ESPAÑHOL COLOQUIAL</DOM>
    <PATTERN>COGE HA CONDUcido</PATTERN>
    <SERVICE>NEGRITO</SERVICE>
  </TA>
</CASE>
<CASE>
  <PATTERN>[SUBSTANTIVO] [ADJETIVO]</PATTERN>
  <TA>
    <DOM>INGLÊS COLOQUIAL</DOM>
    <PATTERN>[ADJETIVO] [SUBSTANTIVO]</PATTERN>
    <SERVICE>DICCIONARIO</SERVICE>
  </TA>
  <TA>
    <DOM>ESPAÑHOL COLOQUIAL</DOM>
    <PATTERN>NULL</PATTERN>
    <SERVICE>DICCIONARIO</SERVICE>
  </TA>
</CASE>

```

Pode-

se observar no segundo caso, o valor NULL no segundo TA. Conforme explicado na seção isso indica que quando selecionado, o TA respeita o padrão original.

### 6.3.2. Eliminador de palavras proibidas

#### Serviços disponíveis:

- **Notificação:** um serviço envia uma mensagem tanto para o usuário do sistema, quanto para o administrador.

**Escopo:** Um sistema que substituí expressões proibidas por “\*\*\*\*” e envia um e-mail que serve como notificação ao administrador do sistema e como advertência ao usuário que utilizou a palavra imprópria.

Esse tipo de aplicação é muito útil em ambientes como fóruns e salas virtuais de bate-papo, onde os usuários são identificados. Logo, será presumido que o serviço de notificação conhece os e-mails dos usuários e do administrador do sistema.

**Elicitação do conhecimento do especialista:** Este é um sistema extremamente simples de se especificar, pois é basicamente uma Memória de Tradução, na qual o novo padrão que irá substituir o reconhecido é sempre “\*\*\*\*”.

**Determinar as ações do Caso Padrão:** Esse é um exemplo de sistema no qual os domínios origem e destino são o mesmo, pois se trata de uma aplicação que realiza filtragens sobre uma mensagem e a outra ação, a que envia os e-mails, não retorna uma saída que precisa ser exibida.



**Formalização dos Casos:** Serão apresentados no quadro 6.2 alguns Casos de como realizar uma substituição simples de padrões.

Quadro 6.2 – Casos de ocultação de palavras proibidas

```

<CASE>
    <PATTERN>VAGABUNDO</PATTERN>
    <TA>
        <DOM>PORTUGUÊS COLOQUIAL</DOM>
        <PATTERN>***</PATTERN>
        <SERVICE>NOTIFICACAO</SERVICE>
    </TA>
</CASE>
<CASE>
    <PATTERN>BICHA</PATTERN>
    <TA>
        <DOM>PORTUGUÊS COLOQUIAL</DOM>
        <PATTERN>***</PATTERN>
        <SERVICE>NOTIFICACAO</SERVICE>
    </TA>
</CASE>
<CASE>
    <PATTERN>RETARDADO</PATTERN>
    <TA>
        <DOM>PORTUGUÊS COLOQUIAL</DOM>
        <PATTERN>***</PATTERN>
        <SERVICE>NOTIFICACAO</SERVICE>
    </TA>
</CASE>
<CASE>
    <PATTERN>CORNO</PATTERN>
    <TA>
        <DOM>PORTUGUÊS COLOQUIAL</DOM>
        <PATTERN>***</PATTERN>
        <SERVICE>NOTIFICACAO</SERVICE>
    </TA>
</CASE>

```

## **7. Considerações Finais**

A TSL mostrou-se interessante por possibilitar de forma fácil e rápida a especificação de sistemas baseados em conhecimento. As pesquisas revelaram métodos que possibilitaram uma convergência entre os tradutores e os demais tipos de aplicações.

Um ponto que pode ser observado em grande parte das aplicações TSL é que inicialmente suas saídas não são muito boas, mas conforme mais fatos são adicionados à especificação, as traduções resultantes tendem a melhorar.

Os problemas e limitações encontrados foram em sua maior parte devido ao protótipo, que não apresenta muitos recursos e um editor para a criação e alteração de especificações TSL se mostrou necessário durante a fase de testes.

De modo geral os resultados obtidos foram positivos, a linguagem é capaz de representar de maneira eficiente o conhecimento, e a capacidade de permitir a utilização de múltiplos sistemas no processamento de um texto permite que a partir de um único texto sejam alimentados vários sistemas computacionais.

Existem diversos problemas que podem ser resolvidos com base na linguagem proposta, a verdadeira dificuldade é explicitar o conhecimento necessário para criar as especificações TSL.

### **7.1. Perspectivas futuras**

O protótipo pode evoluir sendo capaz de realizar análises léxicas, sintáticas e semânticas de forma bem organizada e elaborada. A criação de agentes que possam processar linguagem natural também é algo desejável em versões futuras de interpretadores TSL.

A possibilidade de ter virtualmente qualquer tipo de representação para os conceitos e relacionamentos contidos em uma mensagem trás à tona idéias ousadas como a produção de vídeos a partir de texto e consultas a bases de dados por meio de textos em linguagem natural.

Criar aplicações apenas explicitando o conhecimento torna possível imaginar ambientes permitam a um especialista, especificar sistemas computacionais sem a necessidade de programar sequer uma linha de código.

Em caráter mais imediato, é desejável um editor amigável que possibilite a criação das especificações TSL de forma rápida e fácil. A criação de interpretadores mais inteligentes e elaborados também é uma prioridade, com tratamento de ambigüidades e várias outras melhorias que são necessárias.

## 8. Referências

[Garrão 2001] Garrão, M. de U.; “**Tradução Automática: Ainda um Enigma Multidisciplinar**”; V congresso Nacional de Lingüística e Filologia. Instituto de Letras da UERJ, Rio de Janeiro - 2001.

[Novaes 2006] Amorim, Marcello Novaes de; “**Um Sistema de Gerenciamento de Memória de Tradução**”; Projeto Final; Universidade Federal do Espírito Santo; Departamento de Informática - 2006.

[Assis 1997] Assis, A. P.; “**Tradutores Sintáticos de Textos**”; Projeto de Iniciação Científica; Universidade Federal do Espírito Santo; Departamento de Informática - 1997.

[Martins 2004] Martins, Ronaldo Teixeira; “**A nova língua do imperador**”; Tese de Doutorado; Universidade Estadual de Campinas; Instituto de Estudos da Linguagem - 2004.

[Rino 2002] Rino, Lucia Helena Machado; “**Introdução aos métodos e paradigmas de Tradução Automática**”; NILC-TR-02-04 - Março, 2002.

[Hutchins 2005] Hutchins. J.; “**Example-based machine translation: a review and commentary**”; Machine Translation; Springer – 2005.

[Houaiss 2003] Houaiss, A.; Villar, M.; Franco, FM de Melo; “**Dicionário Houaiss da Língua Portuguesa**”; Objetiva – 2003.

[Aurélio 1988] Ferreira, Aurélio Buarque de Holanda; Anjos, M. dos; “**Dicionário Aurélio básico da língua portuguesa**”; Editora Nova Fronteira - 1988.

[Rino 2001] Rino, Lucia Helena Machado; Martins, Ronaldo Teixeira; Nunes, Maria das Graças Volpe; Jr., Osvaldo Novais de Oliveira; “**O uso de interlíngua para comunicação via Internet: O Projeto UNL/Brasil**”; NILC-TR-01-3 - Julho, 2001.

[Nonaka-Takeuchi 1997] Nonaka; Takeuchi; **“Criação de conhecimento na empresa”**; Rio de Janeiro: Editora Campus – 1997.

[LISA 2007] **The Localization Industry Standards Association**; Disponível em: <http://www.lisa.org/standards/tmx/> ; Acesso em: junho de 2007.

[Martin 2005] Streibel, Martin; **“Implementando Web Services”**; Universidade Federal do Rio Grande do Sul – 2005; Disponível em: <<http://palazzo.pro.br/artigos/martin.htm> >. Acesso em: junho de 2007.

[Almeida 2002] Almeida, M. B.; **“Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares”**; Universidade Federal de Minas Gerais; Mestrado em Ciência da Computação; Disponível em: <[http://www.eci.ufmg.br/mba/text/art\\_xml\\_sub1\\_WEB.pdf](http://www.eci.ufmg.br/mba/text/art_xml_sub1_WEB.pdf)>. Acesso em: junho de 2007.

[S.I.F 2007] **Sistemas Informáticos da Física e Linguagens SGML** ; U.B.I. - Portugal ; <<http://ubista.ubi.pt/~dfis-wg/sgml/sgml.html>>. Acesso em: junho de 2007.

[Souza-Alvarenga 2004] Souza, Renato Rocha; Alvarenga, Lídia; **“A Web Semântica e suas contribuições para a ciência da informação”** ;Universidade Federal de Minas Gerais; Ciência da Informação V. 33, N° 1 - 2004.

[Mendes 1997] Mendes, Raquel Dias. **“Inteligência Artificial: Sistemas Especialistas no Gerenciamento de Informação”**; Ci.Inf., Brasília, v.26, n.1- 1997. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19651997000100006&lng=pt&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19651997000100006&lng=pt&nrm=iso)> . Acesso em: 21 Jan 2007.

[Canuto 2000] Canuto, Anne Magaly de Paula; Howells, Gareth; Fairhurst, Michael; **“Network Combination Strategies for a Hybrid Neur-Fuzzy Multi-Expert System”**. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems”; Madrid – 2000.

**[Menéndez 2002]** Menéndez, Andrés Ignacio Martínez; “**Uma ferramenta de apoio ao desenvolvimento de WebServices**”; Dissertação de Mestrado, Universidade Federal de Campina Grande; Pós-Graduação em Informática – 2002.

**[Breda 2005]** Breda, W.L.; “**Falibras-MT – Um sistema para autoria e uso de tradutores automáticos Português – Libras, baseados em memória de tradução**”; Monografia do Projeto Final de Graduação do Curso de Engenharia de Computação; Universidade Federal do Espírito Santo – 2005.

**[Simões 2004]** Simões, Alberto Manuel; Almeida, José João; Guinovart, Xavier; “**Memórias de tradução distribuídas**”; S.l.:sn ISBN 972-9916608 – 2004.

**[Guia 1979]** Adams, Douglas; “**The Hitchhiker's Guide to the Galaxy**”; Reino Unido; Pan Books – 1979.

**[Gomes 2006]** Gomes, Ramon; Todeschini, Leonardo; “**WEB: Como programar usando ferramentas livres.**”; Editora AltaBooks; Rio de Janeiro – 2006.

**[Kay 1994]** Kay, M.; “**Machine Translation: The Disappointing Past and Present. In Survey of the State of the Art in Human Language Technology.**”; Xerox Palo Alto Research Group; California – 1994.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)