Memórias de Tradução Distribuídas

Alberto Manuel Simões, José João Almeida, and Xavier Gomez Guinovart

Departamento de Informática, Universidade do Minho
{albie@alfarrabio.|jj@}di.uminho.pt

Universidade de Vigo
xgg@uvi.es

Resumo Neste documento apresenta-se o conceito de memórias de tradução distribuídas, a sua utilidade, e como podem ser implementadas usando a tecnologia de *web-services*.

1 Introdução

Na área da tradução assistida por computador usa-se memórias de tradução (MT): correspondências de frases entre duas ou mais línguas diferentes.

$$TMX \equiv \mathcal{S}_{\mathcal{L}_\alpha} \rightharpoonup \mathcal{S}_{\mathcal{L}_\beta} \times \ldots \times \mathcal{S}_{\mathcal{L}_\omega}$$

Os tradutores usam estas memórias como bases de dados onde traduções já efectuadas são armazenadas para que futuras traduções, semelhantes a outras já realizadas sejam reutilizadas.

Para armazenar as MT cada aplicação usa o seu formato proprietário. No entanto existe um *standard* baseado em XML[1] para o intercâmbio das MT: o *Translation Memory eXchange* — TMX[3,2].

Embora cada tradutor construa a sua MT à medida que vai trabalhando nos seus projectos, existem empresas que fornecem as suas memórias de tradução especializadas em determinada área (por exemplo, certas indústrias automóveis) e outras que as vendem (por exemplo, as empresas de software de tradução).

Quando uma destas duas situações ocorre, a TMX é enviada (quer seja por correio, e-mail, ftp, etc) para o tradutor, que passa a ser seu dono[1].

Com a introdução de memórias de tradução distribuídas, onde cada fornecedor as coloca acessíveis usando uma tecnologia de partilha de dados remota, passa a ser possível a "subscrição" ou "aluguer" de memórias de tradução.

No entanto, esta nova filosofia não irá apenas beneficiar o comerciante de memórias de tradução, mas também permitir uma maior facilidade de partilha entre os profissionais da área.

[1] embora em certos casos existam limitações no seu uso — apenas em alguns projectos, durante algum tempo, etc.

2 Arquitectura de Rede

A arquitectura de rede para um sistema de MT distribuídas pode ser construída de duas formas diferentes, como apresentado na figura 1.

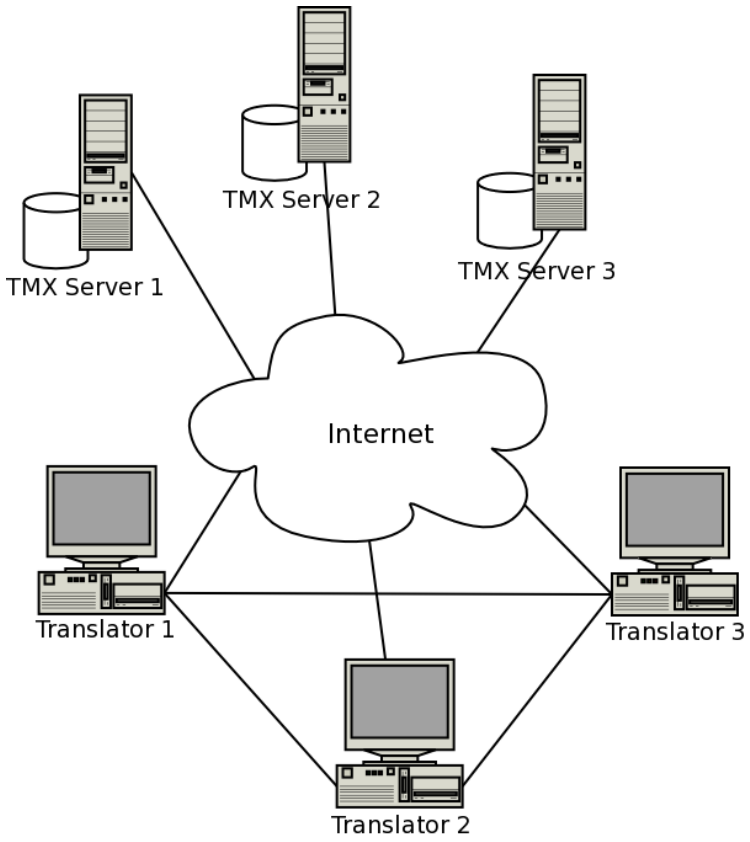

Figura 1. Arquitecturas de rede

Existe a possibilidade de usar uma arquitectura *cliente/servidor* (C/S) ou uma arquitectura *peer-to-peer* (P2P). Cada uma destas arquitecturas tem uma aplicação diferente.

A arquitectura P2P é especialmente útil num gabinete de tradução, onde vários tradutores estão a trabalhar e a partilhar automaticamente as traduções que estão a realizar.

No caso de uma aquitectura C/S, será necessário um servidor dedicado a dispobinibilizar unidades de tradução. Este será o método preferencial para a disponibilização de memórias de tradução por terceiros.

Enquanto que na arquitectura C/S o sistema pode ser implementado com uma qualquer tecnologia baseada em "remote procedure call", quer seja Sun RPC, Corba, Java RMI ou mesmo Web Services.

Em relação ao sistema P2P, uma tecnologia baseada em WebServices não é das mais adequadas já que obrigaria que cada tradutor tivesse um servidor web na sua máquina de trabalho. Neste artigo não nos vamos debruçar sobre os problemas e possíveis soluções para esta arquitectura. Ficam, no entanto, aqui algumas sugestões:

- uso de uma tecnologia proprietária;
- associação a um dos membros da rede P2P uma lista de clientes, que cada membro faz download e comunica usando, por exemplo, sockets;
- criação de um servidor local para submissão de memórias de tradução e posterior divulgação (deixamos de ter arquitectura P2P real passando a C/S);

3 Arquitectura Lógica

Quando usadas localmente, as memórias de tradução são especialmente úteis no momento da tradução: para cada frase ou segmento a traduzir, esta é pesquisada na memória de tradução para ver se foi anteriormente traduzida. Vários softwares realizam este processo de forma ligeiramente diferente, podendo procurar por toda a frase ou apenas por segmentos razoavelmente bem delimitados (por etiquetas HTML, por exemplo, já que muito deste software guarda informação acerca dos comandos usados no documento original).

Em ambiente distribuído o processo será ligeiramente diferente. Além de consultar a sua memória de tradução local, o sistema irá comunicar com um conjunto de servidores TMX, o que poderá resultar em mais do que uma resposta para determinada frase ou segmento, o que obrigará a escolher uma delas.

Esta escolha poderá ser feita de várias formas diferentes:

- baseada na resposta mais rápida — significa que se a tradução já existir na cópia local, o sistema nem chega a comunicar com os sistemas TMX. Caso contrário, o primeiro sevidor que responder será o escolhido;
- baseada na classificação dos servidores TMX — a aplicação cliente define uma classiciação para cada servidor TMX e escolhe, das respostas que recebeu, a que veio do servidor com melhor classificação;
- baseada na classificação das traduções — de acordo com uma proposta para a inclusão de uma medida de qualidade por tradução, o cliente poderia ser capaz de escolher a melhor tradução.

4 Estrutura de uma TMX

Uma memória de tradução segue um DTD que define dois grupos distintos: cabeçalho (`header`) e o corpo (`body`).

O cabeçalho guarda meta-informação como sejam o autor, data de criação, ferramenta que o gerou e outras propriedades. O atributo mais importante para

a disponibilização de memórias de tradução distribuídas é o "`srclang`" que define qual a língua original na criação da memória de tradução (todas as outras línguas são traduções desta língua).

O corpo do documento TMX é composto por uma sequência de unidades de tradução que correspondem a:

```
<tu>
  <tuv xml:lang="en">
    <seg>Configure window properties</seg>
  </tuv>
  <tuv xml:lang="pt">
    <seg>Configurar propriedades janelas</seg>
  </tuv>
</tu>
```

Cada unidade de tradução (`tu`) pode conter texto em mais do que uma língua. Cada um destes textos é colocado num elemento "`seg`" dentro de um outro denominado "`tuv`". Este é identificado obrigatoriamente pelo nome de língua, com o atributo "`xml:lang`".

Os elementos "`tuv`" e "`tu`" permitem um conjunto mais largo de etiquetas mas que iremos referindo ao longo do artigo.

5 Definição do Serviço

Esta secção pretende definir quais os métodos que o servidor TMX deve saber responder para a implementação de um serviço de TMX distribuídas.

Detecção de TMX servidas Para que um cliente possa usar (com sucesso) um servidor TMX é necessário que este tenha TMXs nas línguas desejadas. Não é correcto o uso do serviço para um par de língua inexistentes já que além do uso de largura de banda, estaria a sobrecarregar o servidor.

Com esta ideia em mente, existem duas alternativas para a resolução do problema: perguntar ao servidor sobre determinado par; ou perguntar por todos os pares aos quais o servidor sabe responder.

Qualquer uma das soluções pode ter vantagens. Por um lado, ao perguntar por todos os pares, o cliente pode classificar todos os servidores e não voltar a perguntar a não ser que tenha passado algum tempo. Por outro lado, é provável que o tradutor só traduza um par de línguas, pelo que a pergunta por um par de línguas possa ser mais eficiente.

Acabamos por escolher a primeira alternativa:

$$traduzes? : \mathcal{L}_\alpha \times \mathcal{L}_\beta \longrightarrow 2$$

Pedido de tradução Como é habitual em muitos sistemas cliente/servidor, um servidor de TMX distribuídas terá interesse em ser *state free*. Isto significa que não será possível um cliente indicar que a partir de determinado momento,

todas as *queries* que efectuar serão entre determinado par de línguas. Este tipo de solução implicará que em cada *query* o cliente indique em que língua está a frase que deseja ver traduzida, e para que língua:

$$traduz : \mathcal{L}_\alpha \times \mathcal{L}_\beta \times \mathcal{S}_{\mathcal{L}_\alpha} \longrightarrow \mathcal{S}_{\mathcal{L}_\beta} + 1$$

Referências

1. *eXtended Markup Language (XML) version 1.0 recommendation*. World Wide Web Consortium, 10 February 1998. http://www.w3.org/TR/1998/REC-xml-19980210.html/.
2. OSCAR. Open Standards for Container/Content Allowing Re-use — TMX home page, 2003. `http://www.lisa.org/tmx/`.
3. Yves Savourel. TMX 1.4a Specification. Technical report, Localisation Industry Standards Association, 1997.