



Internacionalização de aplicações Web: uma ferramenta para acelerar a tarefa de tradução em um ambiente de aplicação médica

Internationalization of web applications: a tool to speed up translation in health care

Fabício Ferracioli¹

Maria Angélica de Oliveira Camargo-Brunetto¹

Descritores: XML; Internacionalização; Tradução

RESUMO

Objetivos: Com base nesta necessidade, este trabalho teve como objetivo propor uma estratégia para facilitar a tradução de aplicações web com domínio específico e validá-la através da construção de um módulo de tradução agregado a aplicação SacarWeb. **Métodos:** O módulo trabalha separadamente com as sentenças de domínios de conhecimento específicos, sendo papel dos usuários com conhecimento nessas áreas a tradução de sentenças, o que gera traduções com maior fidelidade. Utilizou-se a linguagem de marcação XML para definição das sentenças em qualquer idioma. **Resultados:** Como resultado obteve-se um módulo de tradução integrado ao SacarWeb, que mostrou-se de fácil utilização, reduziu o tempo de tradução das sentenças específicas da área da saúde, e eliminou a necessidade de conhecimentos fora do escopo do idioma de tradução. **Conclusão:** A estratégia proposta pode ser facilmente adaptada para aplicações Web ainda em fase de projeto com outros domínios de conhecimento.

Descriptors: XML; Internationalization; Translation

ABSTRACT

Objectives: The purpose of this paper was to propose a strategy to facilitate the translation of web applications with specific domains and to validate it by creating a translation module integrated with the SacarWeb application. **Methods:** The module separately works sentences of specific domains of knowledge, and the translation of sentences is provided by users with knowledge in these areas, which improves translation accuracy. XML markup language was used for defining sentences in any language. **Results:** It was created a translation module integrated with SacarWeb, which proven to be easy to use, reduced translation time of specific health expressions, and eliminated the need of knowledge outside the scope of the language translated. **Conclusion:** The proposed strategy can easily be adapted to web applications under development in any knowledge domain.

Autor Correspondente:

Fabício Ferracioli

e-mail: fabricioferracioli@gmail.com

¹ Universidade Estadual de Londrina (UEL), Departamento de Computação. Londrina (PR), Brasil.

Artigo recebido: 03/12/2009

Aprovado: 17/06/2010

INTRODUÇÃO

Em geral, aplicações *Web* tornam-se mais úteis quando acessíveis por um grande número de usuários, sem que o idioma seja uma dificuldade. Devido ao caráter mundial da *Web*, Internacionalização é uma característica cada vez mais buscada nessas aplicações.

Como um *Website* pode ser acessado por qualquer pessoa a partir de qualquer lugar, é interessante adicionar recursos que possam ser acessíveis mundialmente, ou pelo menos em alguns países. Neste contexto, é relevante a introdução de alguns conceitos de internacionalização, conforme de Troyer, que destaca três termos que são definidos pela LISA (*Localization Industry Standards Association*): (i) globalização, (ii) internacionalização e (iii) localização.

A Globalização⁽¹⁾ é o processo de disseminação de algo em diversos países, fazendo o conteúdo aplicável e usável nos mesmos. Internacionalização⁽¹⁾ é um processo de nível técnico que permitirá ao produto ser localizado.

Localização⁽¹⁾ trata da adaptação de um produto a um mercado específico, envolvendo quatro aspectos: lingüístico, físico, negócios e cultura e questões técnicas. A tradução é uma das tarefas que compõem a localização, sendo considerada a mais custosa. Apesar de consumir mais recursos é essencial para globalização, sendo em alguns casos utilizada no mesmo conceito de globalização, internacionalização e localização⁽²⁾.

De acordo com o Miniwats Marketing Group⁽³⁾, em Março de 2009 a quantidade de usuários da Internet no mundo era maior que 1 bilhão e 500 milhões, sendo mais de 80% destes concentrados na América do Norte, Ásia e Europa. Esses números justificam a atenção a ser dada a usuários com idiomas e culturas diferentes das nativas. Algumas razões para justificar a importância de *sites* multi-idioma são⁽⁴⁾: (i) é uma oportunidade para conseguir parcerias em outros países; (ii) de uma perspectiva comercial, é possível para pequenas empresas criar presença global facilmente. Em contrapartida, grandes empresas não parecem ser tão competitivas quando não possuem sites internacionalizados; (iii) de um ponto de vista acadêmico, pode ser uma maneira de atrair estudantes internacionais para uma universidade e disseminar informação para a comunidade acadêmica.

De maneira a oferecer aplicações *Web* compreensíveis, obter mais usuários e também uma maior aceitação deles é particularmente interessante adicionar características culturais além dos multi-idiomas as aplicações⁽⁵⁾.

Visto que a tradução é uma tarefa custosa, existem diferentes estratégias para sua realização. Sistemas automatizados podem ser encontrados em *Content Management Systems* (CMS) e *Frameworks* além das implementações específicas do GNU gettext⁽⁶⁾. Mesmo sendo efetivos em seus objetivos, esses sistemas

possuem limitações em recursos para a tradução. A principal é a necessidade de o tradutor ter conhecimento não apenas do domínio do *site* no idioma desejado, mas também da estrutura utilizada para armazenamento e formatação do texto traduzido, o que pode ser uma dificuldade adicional. Além disto, a edição manual de arquivos é um processo suscetível a erros humanos.

Para evitar este processo de edição manual, uma alternativa viável seria os próprios integrantes da equipe de desenvolvimento de uma aplicação realizar a tarefa de tradução. Esta abordagem apresenta as seguintes desvantagens: o profissional de software é deslocado de suas funções originais; no caso de domínios específicos, pode ocorrer de nem sempre o tradutor ter conhecimento do domínio abordado, possibilitando a ocorrência de graves inconsistências na tradução.

Assim, torna-se necessário a adoção de estratégias e ferramentas que possibilitem a disponibilização de aplicações multi-idiomas de maneira facilitada e consistente. Este artigo apresenta uma estratégia para facilitar a tradução de aplicações *Web* com domínio de conhecimento específico e uma ferramenta de tradução que foi desenvolvida e integrada em uma aplicação *Web* da área médica.

APRESENTAÇÃO DO PROBLEMA

A motivação para o presente trabalho partiu de uma aplicação já existente, o SacarWeb (Software para Avaliação Cardio-Respiratória na Web). Ele possui como um de seus objetivos a construção de uma grande base de dados clínicos compartilhado entre diferentes unidades de saúde, com aspectos culturais e geográficos próprios⁽⁷⁾. A aplicação foi desenvolvida utilizando principalmente a linguagem de programação PHP (PHP: *Hypertext Preprocessor*)⁽⁸⁾ e o idioma Português Brasileiro.

Durante o desenvolvimento do SacarWeb observou-se a necessidade de divulgar a aplicação em outros países, através de cooperação por parcerias. Neste sentido, sempre que possível disponibilizou-se diferentes padrões de interpretação para os exames, específicos para alguns continentes. Essa estratégia faz parte do processo de internacionalização, que foi aplicada em alguns módulos do SacarWeb como o Teste de Função Pulmonar e o Pressões Respiratórias. Os padrões de interpretação utilizados são o Brasileiro, Americano e Europeu⁽⁹⁾.

Apesar desses esforços, o software precisava ser localizado, uma vez que havia interesse na sua utilização por pesquisadores de institutos clínicos de diferentes países. Para isso foi necessário a tarefa de tradução, que demandava bastante trabalho, considerando-se a presença de termos específicos do domínio de conhecimento médico. Mediante esta necessidade, o objetivo desse estudo foi propor uma estratégia de tradução para aplicações Web com domínio de

conhecimento específico, onde seja necessário somente o conhecimento dos idiomas envolvidos na tradução, com baixa curva de aprendizado e que possibilite uma tradução rápida para equipes com membros com conhecimentos de áreas heterogêneas.

MÉTODOS

A estratégia inicial adotada, para localização na aplicação SacarWeb e a estratégia proposta, que visa maior agilidade no processo de tradução, são descritas nesta seção.

Estratégia Inicial para localização do SacarWeb

Inicialmente, foi utilizada uma estratégia recomendada pelo W3C⁽¹⁰⁾: a definição de sentenças utilizadas no software através de arquivos com definições estáticas. No SacarWeb, isso foi realizado utilizando o recurso de constantes da linguagem de programação PHP.

Nessa abordagem foi definido um identificador para cada sentença, assim como seu valor no idioma desejado. Para cada idioma alvo de tradução, gerou-se um arquivo contendo os mesmos identificadores, porém com os valores traduzidos no respectivo idioma. O arquivo apropriado era selecionado de acordo com o idioma preferido do usuário. Essa abordagem apresentou algumas desvantagens: (i)

qualquer edição de sentenças deveria ser realizada diretamente no(s) arquivo(s) do(s) respectivo(s) idioma(s), requerendo do tradutor, conhecimento da sintaxe da linguagem de programação usada nos arquivos de idiomas; (ii) gerenciamento manual das sentenças traduzidas, já que para cada idioma existe um arquivo e qualquer modificação de sentença deveria ser editada em todos eles. Como todas estas tarefas deveriam ser feitas manualmente, esta abordagem poderia se tornar uma grande fonte de erros.

A falta de automatização dificultava, por exemplo, o rastreamento de sentenças não traduzidas, o que podia ocasionar duplicidade de traduções, ou mesmo nos piores casos, erros de sintaxe poderiam comprometer o bom funcionamento da aplicação.

Dois dificuldades foram observadas nesta estratégia: (i) a necessidade de edição dos arquivos de idiomas requer o domínio da sintaxe dos mesmos, o que pode ser uma fonte adicional de erros; (ii) risco de traduções imprecisas de termos da área médica, no caso de serem realizadas por profissionais da área de informática.

Estratégia Proposta

Tendo em vista as dificuldades da abordagem utilizada, foi necessário rever a estratégia de localização. Os objetivos a serem alcançados nesta nova abordagem foram: (i) facilidade no gerenciamento dos

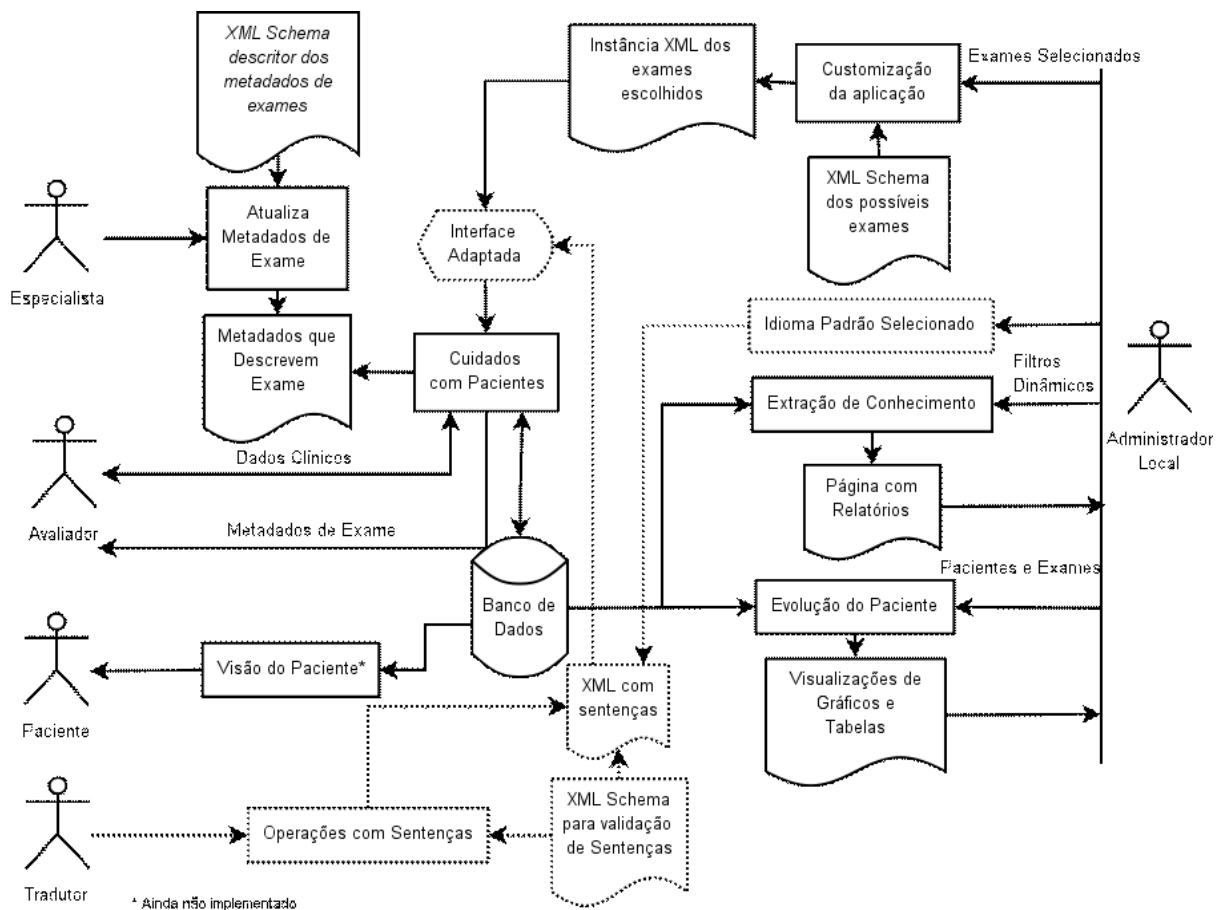


Figura 1 - Arquitetura do SacarWeb

arquivos de sentenças; (ii) colaboração entre os diferentes tipos de especialistas presentes na equipe de desenvolvimento do SacarWeb; (iii) facilidade de utilização por *parte* dos usuários especialistas da saúde; (iv) fidelidade das traduções, principalmente as relacionadas a termos médicos; (v) facilidade de integração com o SacarWeb.

A estratégia proposta foi baseada no fato do SacarWeb apresentar dois tipos de sentenças, as de uso geral da aplicação e as específicas do domínio de conhecimento da saúde. As sentenças gerais normalmente estão presentes em botões de navegação, mensagens de confirmação ou avisos. Já as pertencentes ao domínio da saúde são de extrema importância para o correto entendimento da interpretação dos exames e avaliações disponíveis no SacarWeb. Levando em conta essa separação de sentenças, observou-se que a tradução poderia ser agilizada, considerando-se duas categorias de tradutores: (i) especialistas da área médica e (ii) tradutor com conhecimento da terminologia de aplicações web. A sugestão é que boa parte dos tradutores sejam os próprios usuários do sistema, o que é uma prática encorajada, uma vez que ao participar do processo de desenvolvimento do software, os objetivos dele são atingidos mais rapidamente⁽¹¹⁾.

A estratégia proposta foi implementada através de uma ferramenta de tradução integrada no SacarWeb. Salienta-se que a estratégia proposta, bem como as técnicas usadas na ferramenta podem ser aproveitadas para qualquer aplicação *Web* com domínio de aplicação específico. A Figura 1 exibe a arquitetura do SacarWeb com o módulo de tradução representado pelas linhas pontilhadas. Ela também mostra a interação do módulo e o usuário Tradutor, que realiza a tradução das sentenças que serão armazenadas em um arquivo XML contendo somente um idioma, arquivo que é validado através de um XML Schema. Ao Administrador Local de uma instituição que utiliza o SacarWeb é dada a opção de selecionar o idioma a ser utilizado na aplicação, opção que é resgatada para adaptação da interface para todos os usuários pertencentes aquela instituição.

TRABALHOS RELACIONADOS

Um pacote de internacionalização de *software* muito utilizado é o GNU gettext, que consiste numa série de ferramentas integradas para programadores e tradutores. O conjunto de ferramentas fornece um *framework* que ajuda na produção de mensagens multi-idioma por outros pacotes GNU. Para utilizar o gettext, é necessário ao programador identificar as sentenças que deverão ser traduzidas no código fonte do software e gerar arquivos *Portable Object Template* (POT) que devem atender a uma estrutura específica.

Uma vez que o *template* é criado, o programador deve editar o código fonte do *software*, especificando

o uso do gettext. Após isso, um dos programas que fazem parte do gettext gera um arquivo com as sentenças de origem, com ponteiros para os pontos onde elas deverão ser utilizadas no código fonte e, todas as traduções necessárias até o momento estão vazias. Nesse ponto é gerado o arquivo PO, que contém inicialmente a cópia das sentenças de origem. Esse arquivo será preparado por um programador, para posterior edição por um tradutor. A tarefa de edição dos arquivos PO pode ser facilitada através de editores específicos, como os autores⁽¹²⁻¹³⁾. Uma das grandes vantagens em se utilizar editores são seus recursos adicionais, como verificação ortográfica e sintática, validação do arquivo de entrada, entre outros.

O GNU gettext é altamente utilizado, devido a sua eficiência em lidar com internacionalização. Ele está disponível em diversas linguagens de programação, incluindo o PHP. Bradley⁽¹⁴⁾ exemplifica a utilização do gettext em aplicações *Web* desenvolvidas em PHP.

Outra técnica proposta por Parr⁽¹⁵⁾ é o tratamento do texto como um recurso, separado do código fonte e acessível ao tradutor, de maneira dinâmica. A técnica é utilizada por Bradley⁽¹⁴⁾ com a linguagem PHP. Nessa abordagem, a manutenção da aplicação é facilitada para os programadores. Entretanto, algumas partes expressivas de código HTML são mantidas no PHP, o que é uma característica não desejável devido à separação em camadas de aplicações *Web*. Como solução para esse problema, o autor propõe a utilização do gettext.

Limitações do gettext

No contexto do SacarWeb, o gettext certamente seria uma ferramenta aplicável e efetiva. Entretanto, devido às necessidades levantadas para o módulo de tradução, esta opção foi descartada. Um trecho arquivo PO utilizado na implementação do gettext para PHP pode ser visto no Quadro 1.

A sintaxe usada em um arquivo PO, pode desmotivar um potencial tradutor a participar do processo de tradução de um *software*. Mesmo que o tradutor tenha domínio da sintaxe para editar o arquivo, o processo é bastante suscetível a erros, o que pode ocasionar falhas de difícil detecção na aplicação. A utilização de editores específicos pode ajudar nesse segundo problema, entretanto é necessário realizar o *download* e a configuração do ambiente, que pode diferir entre diferentes tradutores e causar conflitos entre arquivos.

Apesar dessa limitação, com as ações da comunidade de *software* livre, o gettext é extremamente efetivo e utilizado em aplicações de todos os tipos. Como exemplo temos a distribuição de Linux openSuse⁽¹⁶⁾, o gerenciador de janelas KDE⁽¹⁷⁾, o *framework* CakePHP⁽¹⁸⁾, entre outros. Entretanto, em aplicações da área da saúde e outros domínios específicos, a necessidade da edição dos arquivos pode ser um fator limitante. Assim, propôs-se uma

ferramenta de tradução que foi integrada ao SacarWeb.

Quadro 1 - Exemplo de arquivo PO

```
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=ISO-8859-1\n"
"Content-Transfer-Encoding: 8bit\n"
#: RSSHandler.php:31
msgid "News type is not valid."
msgstr "Die Nachrichtkategorie ist falsch."
#: RSSHandler.php:37
msgid "found %d fatal error"
msgid_plural "found %d fatal errors"
msgstr[0] "s'ha trobat %d error fatal"
msgstr[1] "s'han trobat %d errors fatals"
```

Ferramenta de Tradução do SacarWeb

A ferramenta de tradução do SacarWeb é responsável pelo gerenciamento dos idiomas, além da marcação correta do código HTML, uma vez que quanto mais um conteúdo é marcado corretamente com informações sobre idioma, mais útil e disseminado ele será⁽¹⁹⁾. Algumas questões de implementação foram consideradas: (i) tratamento do texto como um recurso, separado do código HTML, ficando assim facilmente acessível ao tradutor⁽¹⁵⁾; (ii) tratamento de textos e informações sobre imagens como dados dinâmicos, de maneira que o tradutor possa trabalhar independentemente; (iii) representação dos dados de forma a facilitar a manipulação e manutenção das sentenças.

O *eXtensible Markup Language* (XML) atendeu os três requisitos anteriores, principalmente pela existência da API DOM, que possui uma grande quantidade de métodos que facilitam a manipulação desse tipo de arquivo, disponível em diversas linguagens de programação.

A ferramenta de tradução de sentenças é dividida em módulos com objetivos secundários. A figura 2 descreve o fluxo de dados entre eles. Os arquivos PHP com as sentenças utilizadas anteriormente foram aproveitados, uma vez que armazenavam todas as sentenças do SacarWeb. A conversão manual desses arquivos para o formato desejado em XML seria

trabalhosa. Para isso, foi desenvolvido um programa em Java que recebe como entrada o arquivo PHP contendo as sentenças do SacarWeb, gerando como saída o documento XML. A estrutura do XML gerado é semelhante ao Quadro 2.

A gramática desse documento é controlada por um XML *Schema*, utilizado para validação da saída do tradutor e das operações a serem realizadas sobre os arquivos posteriormente.

Uma vez que os papéis de tradução foram divididos entre os especialistas da saúde e da computação, os arquivos com sentenças também tiveram essa divisão. São dois os tipos de arquivo: com sentenças de escopo geral e do domínio da saúde. Através da separação entre os tipos de sentenças, em diferentes arquivos e associados a diferentes tipos de usuário, espera-se obter uma melhor consistência na tradução, uma vez que as expressões técnicas do domínio da saúde são reconhecidas mais facilmente por um especialista da área, assim como suas possíveis traduções.

A próxima etapa do fluxo, a divisão do XML gerado pelo programa Java, requer procedimentos manuais que atendam o critério descrito anteriormente. Para isso é preciso uma avaliação semântica e da utilização da sentença dentro do SacarWeb.

Outro processo manual necessário foi a modificação da aplicação, uma vez que a estratégia anterior estava implementada nativamente no PHP. Alterações foram realizadas para utilizar os métodos associados aos arquivos XML. Para facilitar a utilização do recurso, foram desenvolvidos métodos de chamada em uma classe já existente na aplicação, responsável pelas operações com a interface. Essa classe realiza as chamadas à nova classe, criada apenas para manipulação das sentenças. Essa abordagem tornou mais fácil a adaptação uma vez que não foi necessária uma nova instância da classe de tradução em todos os arquivos de manipulação de interface.

A modificação necessária de maior expressividade foi a alteração das chamadas de constantes para chamadas de métodos. Apesar de simples, pois os identificadores das sentenças permaneciam os mesmos, o trabalho levou algum tempo, uma vez que a

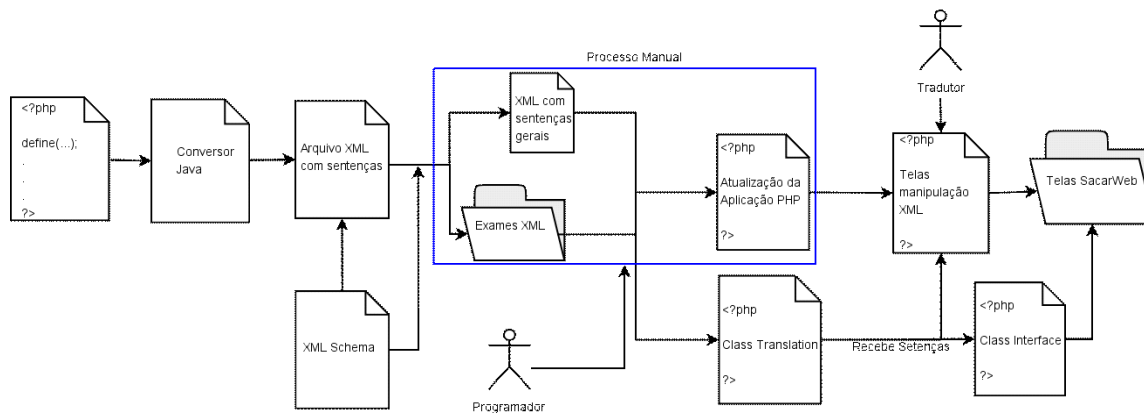


Figura 2 - Fluxo de dados do módulo tradutor

Quadro 2 - Exemplo de XML com sentenças

```

<strings>
  <sentenca xml: id="Sentence ID">
    <traducao xml: lang="pt-BR">Brazilian Portuguese Sentence</traducao>
    <traducao xml: lang="en">English Sentence</traducao>
  </sentenca>
</strings>

```

quantidade de sentenças era elevada. Finalmente, foram implementadas as telas de manipulação de sentenças, utilizadas pelos usuários responsáveis pela tradução.

Ressaltamos que algumas etapas envolvendo procedimentos manuais só foram necessárias devido a existência de código legado oriundo da primeira abordagem multi-idiomas, que impossibilitava a aplicação direta da solução desenvolvida e precisava ser reaproveitado para evitar retrabalho. Numa aplicação em estágio inicial de desenvolvimento seriam necessárias somente as seguintes etapas:

- Obtenção do módulo e schemas XML
- Instalação e configuração do módulo, compreendida basicamente pela definição da localização dos arquivos XML

Após a realização dessas tarefas, o desenvolvimento é realizado de modo natural, utilizando os recursos disponibilizados pelo módulo durante a programação.

RESULTADOS E DISCUSSÃO

Foi implementada para o SacarWeb uma ferramenta de tradução que utiliza XML como forma de representação de dados nos idiomas atendidos por ele. Visando maior fidelidade das traduções, um dos requisitos atendidos foi a possibilidade da participação dos usuários no processo, devido à presença de sentenças específicas da área da saúde.

Após as fases de desenvolvimento e testes, a solução foi colocada em produção. Um usuário especialista da área da saúde ficou responsável pela tradução de 446 sentenças específicas de seu domínio de conhecimento. Essa tarefa foi completada em aproximadamente 4 horas. Esse resultado foi considerado satisfatório, atendendo as expectativas pretendidas com o módulo. Com esse experimento prático, foi possível constatar as seguintes melhorias em relação à abordagem anterior:

- Diminuição do tempo de tradução;
- O conhecimento requerido dos tradutores depende somente do idioma alvo de tradução e do domínio da aplicação;
- Fácil aprendizado, uma vez que as sentenças foram traduzidas rapidamente e pela interface se encontram dentro do próprio SacarWeb, o que evita uma nova curva de aprendizado para uma aplicação diferente;
- Possibilidade de colaboração entre diversos usuários, pois as sentenças são visíveis por todos os usuários tradutores.

Após o término da implementação do módulo de tradução, foi constatada a existência de aplicações com características semelhantes, o que reitera as ideias apresentadas.

No *site* da Mozilla *Foundation*⁽²⁰⁾ são oferecidos recursos para tradução de textos de ajuda dos produtos da empresa. O programa funciona *online*, e a comunidade pode colaborar no processo de tradução. Entretanto, é necessário algum tempo para aprender a sintaxe a ser utilizada para edição de *strings* de formatação, semelhantes às encontradas no *gettext*.

O módulo disponibilizado no SacarWeb é muito semelhante à ferramenta disponível no agregador de projetos *open source launchpad*⁽²¹⁾. Trata-se de uma aplicação bastante expressiva, sendo capaz de oferecer sugestões de possíveis traduções para uma sentença, baseada em outras fornecidas para a mesma *string* em projetos hospedados pelo *site*.

Também foi encontrada uma especificação de marcação do W3C específica para internacionalização de documentos com marcação baseada em XML, chamada *Internationalization Tag Set (ITS)*⁽²²⁾. Através de uma abordagem que utiliza marcação XML, semelhante à desenvolvida no SacarWeb, entretanto com maior poder de tradução, é possível realizar a tarefa de localização de documentos para qualquer idioma. Por ser um meio mais robusto, estar em conformidade com outros padrões do W3C e ser um padrão internacional recomendado pelo órgão, propõe-se como trabalho futuro a utilização da especificação ITS como base dos arquivos de sentenças XML utilizados pelo SacarWeb.

Sendo assim, os resultados alcançados remetem às necessidades levantadas para o desenvolvimento do módulo: facilidade na edição das sentenças, abstração de linguagens de programação e/ou marcação e fácil acoplamento da ferramenta.

CONCLUSÃO

Este artigo apresentou uma estratégia para agilizar a tradução de aplicações Web de domínios de conhecimento específico bem como uma ferramenta embutida numa aplicação médica. A ferramenta já está em uso pelos especialistas da saúde do domínio de conhecimento cardio-respiratório, e o processo de tradução mostrou-se bastante facilitado. As traduções de escopo geral estão sendo realizadas pela equipe de desenvolvimento da aplicação. Até o momento, foram

traduzidas todas as sentenças da área médica, possibilitando rapidamente a obtenção de uma versão localizada para o idioma Inglês do SacarWeb. Devido a isso, acreditamos que quando necessária a localização para outros idiomas, ao se ter uma equipe com

domínio do idioma desejado, rapidamente o objetivo será atingido novamente. A eliminação de barreiras técnicas para tradutores de um determinado domínio, pode agilizar e facilitar a tarefa de tradução de aplicações com domínio de conhecimento específico.

REFERÊNCIAS

1. The Localization Industry Standards Association (LISA). The globalization industry primer. Switzerland; 2007.
2. De Troyer O, Casteleyn S. Designing localized web sites. Proceedings of the 5th International Conference on Web Information Systems Engineering (WISE2004); 2004; Brisbane, Australia: Springer-Verlag; 2004.
3. World Internet Statistics [homepage on the internet]. Miniwatts Marketing Group; c2001-10 [updated 2010 May 27; cited 2010 Jun 11]. Available from: <http://www.internetworldstats.com/stats.htm>.
4. Alvarez MG, Kasday LR, Todd S. How we made the web site international and accessible: A case study. Proceedings of the 4th Conference on Human Factors & the Web. 1998 Jun [cited 2010 Jun 11]. Available from: <http://research.microsoft.com/en-us/um/people/marycz/hfweb98/alvarez/index.htm>.
5. Language on the web [homepage on the internet]. Massachusetts: World Wide Web Consortium; c2009 [updated 2009 May 01; cited 2010 Jun 11]. Available from: <http://www.w3.org/International/getting-started/language>.
6. Gettext – GNU Project [homepage on the internet]. Free Software Foundation; c1998-09 [updated 2010 Jun 06; cited 2010 Jun 11]. Available from: <http://www.gnu.org/software/gettext/>.
7. Rossi ALD, Oliveira Camargo-Brunetto MA. Uma aplicação Web para análise de exames cardio-respiratórios. Anais do V Workshop de Informática Médica. 2005 Jun [cited 2010 Jun 11]. Disponível em: <http://www.ime.uerj.br/professores/cecas/V%20WIM/andamento/8201.pdf>.
8. PHP: Hypertext preprocessor [homepage on the internet]. The PHP Group; c2001-2009 [updated 2010 Jun 11; cited 2010 Jun 11]. Available from: <http://www.php.net>.
9. Amorim TA, Oliveira Camargo-Brunetto MA, Santos KD, Ferracioli F. Remodelagem do software SacarWeb usando técnicas de engenharia reversa e reengenharia de software. Anais do CBIS. 2006;128;133-6. Disponível em: <http://www.sbis.org.br/cbis/arquivos/1030.pdf>.
10. World Wide Web Consortium (W3C). W3C; c2009 [cited 2010 Jun 11]. W3C. Available from: <http://www.w3.org>.
11. Hulse NC, Rocha RA, Fiol GD, Bradshaw RL, Hanna TP, and Roemer LK. Kat: A ûxible XML-based knowledge authoring environment. J Am Med Inform Assoc. 2005;418:430-2.
12. KBabel – KDE Brasil [homepage on the internet]. KDE e V; [cited 2010 Jun 11]. KBabel. Available from: <http://br.kde.org/kbabel>.
13. Poedit [homepage on the internet]. Vaclav Slavic; [cited 2010 Jun 11]. Available from: <http://www.poedit.net/>.
14. How to internationalize your php apps. International Business Machines Corp; c1994-10 [cited 2010 Jun 11]. Available from: <http://www.ibm.com/developerworks/library/os-php-intl/>.
15. Parr T. Web application internationalization and localization in action. ICWE. 2006: Proceedings of the 6th international conference on Web engineering; 2005;64:70-6.
16. Welcome to open SUSE.org [homepage on the internet]. Novell Inc; c2008. Localize It. Available from: <http://i18n.opensuse.org>.
17. KDE Localization [homepage on the internet]. KDE e V; [cited on 2010 Jun 11]. Available from: <http://i18n.kde.org>.
18. Cake PHP: the rapid development PHP framework [homepage on the internet]. Cake Software Foundation Inc.; c2005-2010 [cited 2010 Jun 11]. Available from: <http://cakephp.org/>.
19. Ishida R. Internationalization best practices: Specifying language in XHTML & HTML content. W3C Working Group Note [cited 2010 Jun 11]. Available from: <http://www.w3.org/TR/i18n-html-tech-lang/>.
20. Suporte do Firefox [homepage on the internet]. Mozilla Foundation; c1998-2010 [cited 2010 Jun 11]. Available from: <http://support.mozilla.com>.
21. Launchpad translations [homepage on the internet]. Canonical Ltd.; c2004-2010 [cited 2010 Jun 11]. Available from: <https://translations.launchpad.net/>.
22. Lieske C, Sasaki F. Internationalization tag set (ITS) version 1.0. W3C Recommendation [cited 2010 Jun 11]. Available from: <http://www.w3.org/TR/its/>.