

INTRODUÇÃO À INTERNACIONALIZAÇÃO E À LOCALIZAÇÃO DE SOFTWARE

Achilles Colombo Prudêncio
Departamento de Informática e Estatística/UFSC
achilles@geness.ufsc.br

Djali Avelino Valois
Departamento de Informática e Estatística/UFSC
djali@geness.ufsc.br

José Eduardo De Lucca
Departamento de Informática e Estatística/UFSC
delucca@inf.ufsc.br

Resumo: O objetivo deste artigo é apresentar de uma maneira simples e introdutória o que significa Internacionalização e Localização de Software, mostrando os conceitos correlatos e como se dá o processo de internacionalização e localização no desenvolvimento de um programa. Pretende-se demonstrar a importância da aplicação destes conceitos e técnicas quando se deseja atingir mercados externos e proporcionar melhor qualidade de software e serviços. Apresentam-se os principais papéis dos profissionais da tecnologia da informação e da tradução dentro deste nicho de mercado e também se introduzem alguns conceitos técnicos básicos das principais tecnologias utilizadas, visando apresentar esse mundo ao leitor.

Palavras-chaves: Internalização e Localização de Software; Aplicação de Software à tradução; Aprimoramento de Software e Serviços.

Abstract: The purpose of this article is to present the meaning of Software Internationalization and Localization in a simple and introductory manner. Related concepts are explained as well as the internationalization and localization process in the development of a program. The article demonstrates the importance of the application of these concepts and techniques when there is a desire to reach foreign markets and offer better quality software

and services. The principal roles of Information Technology and Translation professionals within this market niche are presented. The paper also introduces some basic technical concepts of the principal technologies utilized, to help the reader better understand this world.

Keywords: Software Internationalization and Localization; Computer Aided Translation; Software and Services Enhancement.

Introdução

Com a globalização da economia e o advento da Internet, tornou-se muito mais fácil atingir os mercados externos. Quem usa o computador diariamente, já notou que a maioria dos programas usados hoje em dia não foi criada no Brasil, embora muitos deles estejam em português. A globalização e a conseqüente interação de mercados fizeram com que a Internacionalização e a Localização de Software e de websites passassem a ser uma preocupação constante para as empresas que agem ou pretendem agir globalmente.

Para exportar software, assim como para exportar qualquer outro produto, é necessário atender a certos requisitos de qualidade. Alguns países inclusive exigem por lei que o software vindo de fora seja traduzido para o seu idioma, como os países bálticos (ALBUQUERQUE). Para tal, é necessário que se pense desde o princípio do desenvolvimento do programa que este será distribuído em outros idiomas.

Todo o texto do programa visível pelo usuário deve poder ser traduzido, de forma a adaptar-se ao novo local de distribuição. Mais do que isso, o texto do programa deve ser *culturalmente neutro*, ou seja, não deve utilizar expressões que fazem parte de uma determinada cultura específica (como gírias e expressões idiomáticas, por exemplo). O mesmo deve ocorrer com quaisquer imagens que façam parte da interface do programa, como os ícones.

Também se devem levar em consideração diversos outros aspectos culturais e de convenção, dos quais o idioma é somente o

mais perceptível. Outros aspectos são a direção da leitura / escrita do texto, regras de arredondamento dos números, a representação de números em geral, ordenação alfabética, formatos de data, hora, endereços e calendários. Por exemplo, nos Estados Unidos, na representação de data, o mês vem antes do dia, diferentemente da representação usada no Brasil. Além disso, há países que utilizam outro calendário que não o Gregoriano, adotado na maioria, mas não na totalidade dos países. Quanto ao formato de hora, no Brasil adota-se o formato de 24 horas, enquanto nos Estados Unidos utiliza-se 12 horas com indicação AM/PM para diferenciar os dois períodos do dia.

Outros pontos relevantes são os métodos de ordenação de dados. É necessário decidir se uma letra *a* acentuada deve aparecer antes ou depois de uma letra *a* craseada ou com circunflexo, por exemplo. As regras de ordenação mudam de idioma para idioma e de país para país (mesmo dentro de países com o mesmo idioma). Também as técnicas de leitura de teclado de caracteres utilizados em outros idiomas (como o ñ do espanhol ou caracteres cirílicos, árabes, japoneses, etc) e não presentes no idioma de origem do software agregam complexidade aos sistemas.

É necessário levar em consideração todos estes detalhes (e muitos outros) para desenvolver um programa de qualidade e que possa ultrapassar fronteiras. As técnicas de Internacionalização e Localização de Software permitem uma abordagem sistemática e objetiva destes problemas.

A Internacionalização (I18N, no jargão da área¹) é uma etapa do processo de desenvolvimento do software que deve fazê-lo flexível e neutro em termos de relações culturais, financeiras e legais de um país. Um software internacional deve admitir, por exemplo, distintas formatações de números e algoritmos de ordenação que sigam as diferentes regras dos diversos idiomas. Para implementar estas características é necessário utilizar procedimentos adequados, para que seja possível localizar o software apropriadamente. A Localização (L10N²) é o segundo passo na preparação de um software para o mercado internacional, é neste momento em que

se trata da adaptação do mesmo para um local específico em que ele será utilizado. Envolve tradução, adaptação cultural, adaptação a normas e características do mercado-alvo.

A Internacionalização representa uma mudança no modelo de planejamento e implementação de software que quebra velhos hábitos, remove conceitos culturais, introduz flexibilidade e gera um código adequado aos requisitos dos mais diferentes usuários. As questões tratadas vão desde os caracteres que aparecem na tela, como processá-los (ordená-los, por exemplo), como lê-los do teclado (por exemplo, introduzir caracteres de idiomas asiáticos a partir de teclados ocidentais), fuso horário e formatos padronizados da cultura onde o programa será introduzido e um longo etc.

O trabalho de Localização diz mais respeito a um profissional com perfil de tradutor, com grandes conhecimentos na área de Informática (e possivelmente em um setor de aplicação específico, como software básico, software para engenharia, software de gestão, financeiro, de automação, etc). Acima de tudo, um perfeito domínio do idioma-alvo é fundamental, pois a interface e a documentação do programa (juntamente com o website da empresa produtora) são as faces mais visíveis do produto para o cliente. Por este motivo, nativos do idioma desejado devem sempre realizar a Localização. Esta é uma regra de ouro no mundo da Localização.

Ambas as atividades são complementares e é fundamental que sejam realizadas por profissionais competentes com know-how adequado, para o sucesso de qualquer empreendimento além-mar.

Localização de software

Consultando o dicionário Merriam-Webster³, eis a definição que consta para o termo. Localization: “to make local: orient locally”.

Dessa definição extraímos o propósito da Localização de software nos dias de hoje: tornar um produto, um software, e tudo relacionado a ele, local, adaptado à cultura, ao idioma e às conven-

ções locais. Isso é complementado pela definição dada pela LISA (Localization Industry Standards Association - Associação de Padronização do Setor da Localização): “Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold.” (LOMMEL)

Localização é o processo que adapta o produto ao mercado local. Uma primeira concepção é que é somente o trabalho de tradução dos textos, afinal, realiza-se a Localização a partir do idioma (e aspectos culturais) original de criação do software (inglês, por exemplo), chamado de idioma de origem ou de partida (*source language*), para o idioma local (português, por exemplo), ou idioma-alvo ou de destino (*target language*). Mas também é necessário localizar diversos outros aspectos dos softwares: por vezes, um software envolve, por exemplo, questões legais ou de praxis contábil ou de negócios que não são adequadas ao país de destino. As adaptações que se fazem necessárias também correspondem ao conjunto de atividades que a localização engloba.

Retornando à definição de Localização feita pela LISA, temos mais um conceito importante que se deve esclarecer: o conceito de locale. Voltando ao dicionário Merriam-Webster, encontramos um conceito de locale que se aproxima bastante da definição dada pelo setor de software: “Um lugar ou localidade, especialmente quando observado em relação a um evento ou característica em particular”. Essa é exatamente a visão que se deve ter de um locale quando se fala de Localização de software: uma região específica, com todas as suas nuances culturais, ou pelo menos aquelas que deverão fazer parte e afetar o software.

Características já apresentadas anteriormente: a representação dos números, a exibição dos caracteres ao usuário. Por exemplo: em árabe, todo o texto é escrito da direita para a esquerda, mas os números da esquerda para a direita. Quando se representa estas peculiaridades em um software, usa-se um recurso chamado de escrita bidirecional. Quando se representa um texto em chinês ou japonês, é necessário utilizar caracteres especiais, que fazem parte de fontes capazes de representar dezenas de milhares de caracteres (ou ideogramas) dife-

rentes. É preciso ter em mente que cada locale exige um tipo diferente de recurso, diferente de outros que se usam normalmente.

Não se pode esquecer que, além do idioma, outros aspectos também compõem um locale. Pode-se falar o mesmo idioma em dezenas de países diferentes que usam outros calendários, sistemas de numeração, pesos e medidas e sistemas monetários. O locale *português do Brasil* (caracterizado pela sigla pt-BR) usa o calendário gregoriano, sistema de pesos e medidas Internacional e como moeda temos atualmente o Real, mas no português de Portugal (pt-PT) a moeda é o Euro. O locale inglês canadense (en-CA) adota pesos e medidas do sistema Internacional, mas o inglês da Grã-Bretanha (en-GB) usa o sistema métrico Imperial. Por essas e outras características, os locales, no setor de software, são representados por pares compostos por *língua-região*.

Nem sempre foi assim: houve um tempo em que o locale do país onde a língua se originou era chamado de *padrão* e os outros locales, que falavam a mesma língua mas tinham outras diferenças, recebiam o país como derivação. Algo como *Português Padrão* (de Portugal) e *Português do Brasil*. Por uma questão de diplomacia abandonou-se esta classificação (CID).

Essa abordagem parece contraditória ao que se possa imaginar sobre a internacionalização de produtos. A idéia não seria adaptar um produto a ser vendido globalmente? Por que é necessário então, regionalizá-lo? Na realidade, um processo é complementar ao outro: a Internacionalização (que será definida em detalhes mais adiante nesse texto) é o processo que *generaliza* um software culturalmente, preparando o terreno e facilitando a Localização, que é realizada posterior e separadamente.

Visão geral sobre o processo de Localização de software

A Localização é a adaptação de um produto a todas as particularidades concernentes ao local em que será comercializado – le-

gislação local, identidades culturais, idioma, regras de arredondamento e de ordenação, no caso de software – de tal forma que pareça ter sido produzido no próprio mercado-alvo.

No caso da Localização de software, dividir o processo em etapas favorece o gerenciamento e torna esta complexa tarefa administrável. Após a análise do material original deve-se realizar o levantamento terminológico dos termos utilizados na interface com o usuário e na documentação. É ideal que o cliente (o produtor do software) aprove os termos traduzidos. Em seguida, parte-se para a localização da interface, dos menus, botões, caixas de diálogo e outras strings (indicadores de status, mensagens genéricas e mensagens de erro). Então, parte-se para as traduções da ajuda on-line e da documentação, com base na terminologia pré-aprovada já utilizada na interface. Em seguida, ocorre a revisão técnica e lingüística do material traduzido. Assim, podem ter início a reengenharia, que consiste em redimensionamento dos elementos da interface de usuário, definição de teclas de atalho únicas e compilação dos arquivos com textos localizados. Em seguida, pode-se efetuar a captura de telas já localizadas – para exemplificar a execução de uma tarefa, oferecer contexto ou descrever o resultado de um processamento – para utilização na documentação e ajuda.

Posteriormente, ocorrem os testes e o DTP (*desktop publishing*, editoração eletrônica) da documentação e da ajuda on-line. Finalmente, chegamos ao controle de qualidade (conhecido como *quality assurance* ou QA) pré-entrega. O controle lingüístico envolve um check-list de dezenas de itens, dentre eles “Os nomes de títulos, capítulos, cabeçalhos sempre foram traduzidos da mesma forma?”, “Acrescentaram-se os códigos de discagem internacional aos números telefônicos de contato ou estes foram substituídos pelos números locais?” e “As referências culturais e/ou exemplos foram adaptadas (José da Silva/John Doe)?”. Depois, há o teste de funcionalidade e de compatibilidade do software internacional, envolvendo questões como a entrada e saída de caracteres estrangeiros em teclados diferentes, avaliação do funcionamento adequado de funções como ordenação, busca, separação silábica no idioma-alvo,

integração com o sistema operacional subjacente utilizado no mercado-alvo e outras.

O processo de localização em si supõe um fluxo de trabalho complexo e cuidadoso, para a obtenção de um produto final de qualidade. Em função da complexidade, a localização é quase sempre terceirizada para empresas especializadas, pois se distancia das funções básicas das empresas de desenvolvimento de software e não é uma atividade realizada diariamente para justificar uma estrutura interna fixa para tal. Mas não se pode descuidar em nenhum momento desta atividade, uma vez que a interface e a funcionalidade adequadas do produto representam a diferença entre conquistar um mercado ou demonstrar pouca consideração pelo usuário daquele mercado.

Perfil de um localizador de software

O trabalho envolvido na atividade de Internacionalização e Localização de software exige o consórcio de diversos profissionais com perfis específicos e que ainda não estão amplamente disponíveis no mercado de trabalho brasileiro. Linguístas/terminologistas, tradutores, especialistas em redação técnica, engenheiros de internacionalização, gerentes de projetos de localização e engenheiros de QA são exemplos de novos profissionais que se envolvem nestas atividades. Além disso, os profissionais de desenvolvimento de software necessitam de capacitação adequada ou reciclagem para incorporar as novas técnicas de desenvolvimento.

O localizador atuará na Localização de software e materiais correlatos. É um especialista em tradução na área de software (e, em alguns casos, especialista em software de determinados mercados verticais), com sólidos conhecimentos do idioma em que o software está escrito (idioma de partida) e perfeito domínio do idioma para o qual está traduzindo (idioma-alvo). O idioma-alvo é o idioma materno do localizador.

O localizador deve ser o responsável pela tradução de *conteúdo eletrônico* (websites, textos de ajuda on-line de software, a interface de software – botões, menus, janelas), da documentação técnica de software e material colateral (material de marketing, embalagens de produtos, folhetos de referência rápida, documentação de registro, treinamento baseado em computador, etc), inclusive gráficos e multimídia. É de sua responsabilidade também definir (em cooperação com o produtor do software), traduzir e gerenciar a terminologia envolvida nas atividades de tradução. O foco do trabalho do localizador é a busca por qualidade, eficiência e precisão nas tarefas, coerência terminológica, redução de custos totais, com reuso de traduções legadas.

Também se podem atribuir ao localizador as atividades complementares ligadas às questões lingüísticas, como a redação técnica de manuais, revisão dos textos traduzidos (GUI – interfaces gráficas, ajuda, manuais), levantamento e tradução da terminologia utilizada no software, criação/manutenção de glossários de termos técnicos, etc.

Para que possa desempenhar satisfatoriamente suas funções, o localizador deve atender a determinadas demandas, listadas abaixo:

- Perfeito conhecimento da língua materna (língua-alvo).
- Sólidos conhecimentos da língua estrangeira de partida.
- Conhecimentos de informática.
- Conhecimentos básicos do funcionamento de sistemas operacionais.
- Conhecimentos básicos do funcionamento de gerenciamento de arquivos.
- Conhecer a terminologia e o jargão da área de informática, conforme utilizado no país da língua-alvo.
- Conhecer ferramentas de tradução assistida (bancos de ter-

minologia, memória de tradução, tradução automática, etc) e recursos disponíveis (desde dicionários em papel a dicionários on-line, websites, fóruns de debate, e outras fontes).

- Conhecer formatos de arquivos de dados e saber manipulá-los.
- Conhecer o processo básico de Internacionalização e Localização de Software.

Além disso, com a especialização cada vez maior destes profissionais, torna-se desejável conhecer também outras tecnologias e temas, como:

- Tecnologia subjacente à Web (servidores, navegadores, HTML, XML, etc).
- Conceituação de codificação e conjuntos de caracteres.
- Funcionamento básico de redes de computadores (usuário).
- Programação (linguagens de script e/ou de programação, arquivos de recursos, strings e seus problemas associados, interfaces gráficas, codificação de caracteres).

É ainda muito desejável possuir formação na área de tradução e conhecer os glossários de empresas de software. Ter traduzido textos técnicos (artigos, livros, revistas) da área de informática é fundamental para criar uma base de conhecimento sólida que ofereça suporte ao trabalho de tradução/localização de software e websites.

O desenvolvimento de software no Brasil com vistas ao mercado internacional cria também uma demanda natural pelo serviço de localização, mas com uma peculiaridade em relação ao trabalho que se realiza na maior parte do mundo: os softwares aqui desenvolvidos deverão ser traduzidos do português para idiomas estrangeiros (vertidos, no jargão da tradução). É importante perceber que esta versão de um software escrito em português para

outros idiomas não é uma atividade corrente no setor da localização (até pela falta de tradição exportadora do setor no Brasil).

Outra questão fundamental é a intransigência do mercado internacional quanto à qualidade da interface dos programas (em alguns países, a exigência da adaptação do software ao país é tema de lei). Este é um ponto crucial: a tradução de qualidade é fruto do trabalho de tradutores cujo idioma de destino seja o seu próprio idioma materno. Nas traduções do português para outros idiomas, é necessário que a tradução seja sempre feita por tradutores nativos do idioma de destino e da região para a qual se destina o produto. Em vista desta exigência internacional de qualidade, a localização dos produtos brasileiros deverá envolver profissionais de localização dos distintos países-alvo dos softwares, sob pena de não-aceitação dos produtos naqueles mercados. Afinal, a interface do sistema e a documentação do mesmo são a cara do produto. Assim, se uma interface estiver mal traduzida, a conclusão óbvia é que o produto é ruim e o usuário terá a sensação de descaso do produtor do software para com seus clientes. Pior ainda: se a interface (e/ou a documentação) não estiver traduzida, o produto será simplesmente inaceitável.

Exemplo de ferramenta de apoio à tradução

O trabalho de um tradutor-localizador tem peculiaridades próprias da especialização. Diversas ferramentas computacionais estão disponíveis para garantir qualidade, coerência e produtividade do trabalho do tradutor-localizador. A título de exemplo, pode-se ver a seguir a captura de tela de um software de localização de programas feitos em Java. Esse software, apesar de ser um pouco antigo, é um bom exemplo pois sua interface é simples. Quase todos os programas de localização têm em sua interface os elementos contidos nesse programa.

Cabe ressaltar que nos programas escritos em Java, quando devidamente internacionalizados, todos os elementos de sua interface que são passíveis de tradução estão separados em arquivos próprios.

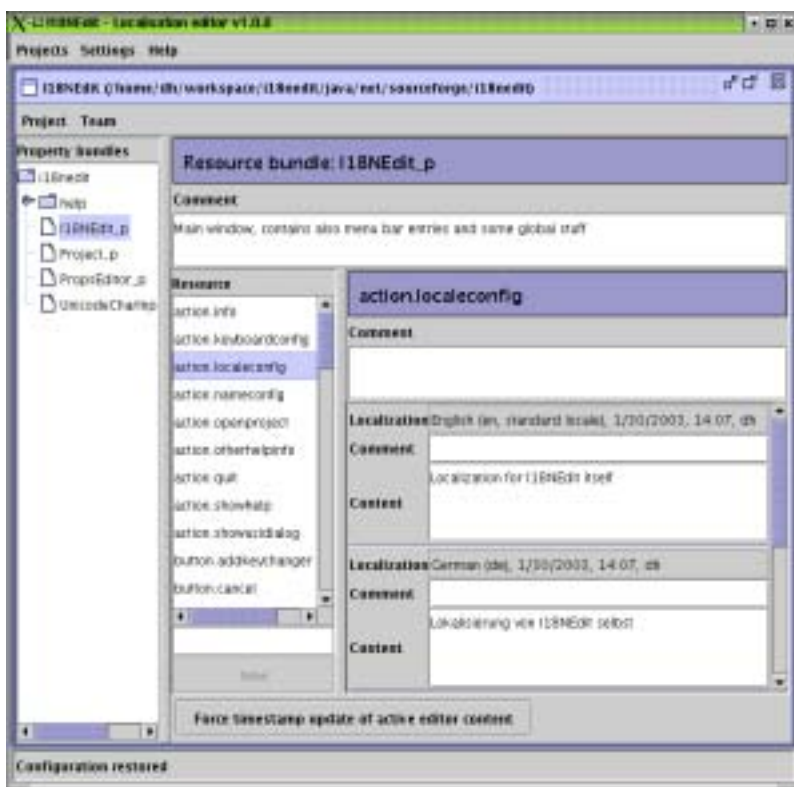


Figura 1 - I18NEdit: ferramenta de auxílio à localização em Java

Na coluna à esquerda da figura, pode-se ver a lista de arquivos com textos para tradução que compõem o programa a ser localizado. No centro, vê-se os vários elementos de interface (Resource), nos quais se encontram os textos a serem traduzidos (como botões, menus, janelas, etc). No canto direito da figura está a tela de edição do texto, que mostra o texto no idioma original e espaço para inclusão da tradução do mesmo.

Em um programa de auxílio à tradução, basicamente esses são os elementos: um editor de textos simples integrado a uma interface que apresenta os arquivos passíveis de localização. A evolução das tecnologias de localização integrou a esse conjunto um banco de dados de termos e/ou orações, que será descrito em detalhes mais adiante. Esse banco de dados assegura a coerência dos termos usados, para que se traduza um mesmo termo ou oração de modo igual à sua tradução anterior.

Problemas do software não internacionalizado

Antes de definir o que é Internacionalização de software, talvez seja necessário mostrar sua importância. Embora seja possível localizar um software sem sua internacionalização anterior, quando isto ocorre, o processo fica extremamente propenso a erros e seu custo final é muito elevado. Pior ainda, ele se restringe apenas à interface visível do programa, desprezando as funcionalidades do mesmo, o que pode inclusive gerar execução incorreta (como a ordenação incorreta de dados, a leitura e exibição incorreta de caracteres, etc).

Não é possível alterar o código executável de um programa, de modo que ele passe a trabalhar com as regras e convenções de outros locais (por exemplo, usar vírgula para separar as casas decimais ao invés de ponto, como ocorre em softwares feitos nos Estados Unidos). Esse tipo de mudança deve ser introduzida diretamente no código-fonte do programa. Escreve-se um programa em determinada linguagem de programação (como Java, C e C++) e chama-se o resultado do trabalho do programador de código-fonte. Este código, escrito na linguagem de programação escolhida, passa por um processo de compilação, que o transforma em um programa executável, que já não é mais legível pelo programador, pois está em uma linguagem de máquina (código-objeto ou código executável). Nos programas comerciais, o usuário comum não tem acesso ao código-fonte de um programa.

Antes da criação das técnicas de internacionalização, uma das *soluções* que a empresa que desejava exportar seu programa para outros mercados adotava era a contratação de uma equipe independente da de desenvolvimento e cujo propósito de trabalho era traduzir todos os textos que fossem necessários dentro do código do programa (ALBUQUERQUE). Desse modo, não se extraía os textos de dentro do código. Após a tradução, compilava-se o programa novamente e assim se obtinha uma nova versão deste, agora *localizada*. Apesar dessa solução ter sido viável a princípio, ela acarreta muitos problemas.

O primeiro deles é que o processo de tradução é muito mais complicado, pois o tradutor precisa ter conhecimentos técnicos de programação, para identificar o que se pode e o que não se pode traduzir dentro do código-fonte do programa. Um descuido ou falta de informação pode introduzir erros no programa, que na pior das hipóteses só serão descobertos na fase final de testes, o que provocará erros de execução em um programa que já deveria estar em uso sem problemas. Por outro lado, se um programador realizar a tradução, a probabilidade de introdução de erros no código diminuirá, assim como a qualidade da tradução.

Outro problema é o fato de que as novas versões localizadas do software tornam a manutenção e atualização muito mais complexas (ALBUQUERQUE). É necessário repassar cada nova atualização ou correção, por mínima que seja, à equipe de localização, que após a conclusão do trabalho devolve a versão localizada à equipe de desenvolvimento, que verifica e corrige novamente problemas do código. Esse processo torna o controle de versões muito mais complicado e demorado, uma vez que é preciso manter controle não só da versão principal do produto, mas de todas as versões localizadas, de forma a assegurar a uniformidade de recursos e funcionalidades de todas as versões. A complexidade do controle de erros cresce muito, pois cada versão pode apresentar erros únicos que exigem registro de acordo com a versão do software.

Conforme o apresentado, a localização sem a prévia internacionalização acarreta vários problemas que tornam inviável a manutenção de um projeto de software, pois não cobre todos os aspectos necessários e introduz uma complexidade desnecessária na etapa de desenvolvimento.

Internacionalização de software

A LISA define **Internacionalização** do seguinte modo: “Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for redesign. Internationalization takes place at the level of program design and document development”. (LOMMEL)

Os conceitos e aplicações da Internacionalização de software surgiram quando o mundo virtual começou a se interconectar, em grande parte devido ao surgimento da Internet. Os bons programas, mais do que realizar sua função, adaptavam-se ao usuário, em termos de língua e convenções culturais. Por exemplo: um software de assessoria contábil devia e deve atender aos princípios e conceitos da Contabilidade aplicada na região de sua utilização (LOMMEL).

Como já visto anteriormente, pegar um software *pronto*, desenvolvido sem qualquer interesse prévio em localização, e simplesmente traduzir todos os textos que aparecerão, em meio ao código-fonte, para o usuário após a compilação do programa, não garante que o produto adapte-se completamente à região em que será vendido. Pior do que isso, pode ser necessário desenvolver o software outra vez, às vezes desde o começo, com pouco ou nenhum reaproveitamento do código anterior. E para localizar esse software para outras regiões será necessário repetir todo o processo.

As técnicas de internacionalização visam separar todo ou quase todo o esforço de tradução do esforço de desenvolvimento do software. Extraem-se do programa todos os textos a que os usuários têm acesso, substitui-se cada texto de dentro do código por uma

palavra única, chamada identificador. Cria-se um arquivo com todos estes textos, seus identificadores e traduções e introduz-se uma função especial no programa, que lê o identificador e o locale determinado no momento de execução do programa e procura imediatamente no arquivo o texto correto a ser exibido. Há também a criação de funções especiais que lidam com outras representações específicas do locale: números, data e hora, ordenação de caracteres e outras.

O fato de todas as strings (seqüências de caracteres que formam orações ou frases) da interface visível ficarem em arquivos separados do código-fonte do programa torna todas as ações mais fáceis, tanto para os programadores quanto para a equipe de tradução que realizará a Localização do software. Os programadores não precisam preocupar-se com estes textos nem com suas traduções, só precisam ter em mente que cada identificador representa um texto que será substituído automaticamente pela função de tratamento.

Os arquivos que contêm o texto traduzível são padronizados. Por exemplo, é possível separá-los em tabelas: na primeira coluna fica o identificador que faz referência ao texto e nas colunas adjacentes ficam o texto na língua original e suas traduções. As funções de tratamento dos programas simplesmente buscam o texto a ser exibido na linha do identificador e na coluna do locale desejado.

Com a padronização do formato destes arquivos, é possível inclusive criar programas para lê-los e editá-los apropriadamente, as memórias de tradução.

Usa-se todo este conjunto de funções, programas e outros recursos para assegurar o máximo de reaproveitamento de código e funcionalidades. Uma vez internacionalizado, é possível localizar um programa tantas vezes quanto for necessário e para tantos locales quanto se desejar.

Além do software

Internacionalizar um programa não se resume a facilitar a tradução de sua interface e ter o cuidado de que ele trabalhe dentro de seu domínio e de acordo com as regras do locale. Existem ainda mais alguns aspectos e elementos visuais da interface que se devem levar em consideração para assegurar o sucesso do produto no mercado.

Além da interface original do software, é imprescindível que toda a documentação que o acompanha – manuais, ajuda, websites, embalagens – seja precisa, clara, sem gírias nem jargões técnicos e completamente neutra e livre de referências e exemplos específicos de uma cultura. (ESSELINK). Por exemplo, traduzir uma frase subjetiva muito significativa no Brasil, algo que “não é uma Brastemp”, perde completamente o sentido no idioma de destino, ou ainda pior, pode adquirir um sentido ofensivo.

Também é preciso tomar esse cuidado ao usar ícones e imagens em botões e outros elementos de interface. O uso de símbolos ou imagens específicos da cultura brasileira, a menos que o contexto onde eles aparecem no programa exija, pode confundir os tradutores quando da localização do programa. O primeiro ícone de lixeira usado no sistema operacional Mac OS, da Macintosh, foi confundido como um ícone de caixa de correio pelos usuários na Europa. (ESSELINK)

A documentação do software deve seguir os mesmos critérios. Geralmente a interface do software contém palavras soltas (nos menus, textos dos ícones) ou frases curtas (de instruções em janelas). Quando se trata da documentação é diferente, nestes casos sugere-se o uso de um recurso lingüístico chamado Linguagem Controlada.

Uma linguagem controlada é um subconjunto da linguagem natural que possui uma quantidade limitada de termos e expressões que se podem usar (ESSELINK). Utiliza-se esse tipo de linguagem principalmente para redigir documentos, manuais e afins. Ele possui apenas um conjunto de termos aprovados pelos responsáveis pela confecção do documento e como a lista de termos depende do domínio do documento, na realidade não existe um padrão para a

Linguagem Controlada. Seu uso assegura que o texto não possua ambigüidades e que só haja uma forma de interpretar cada informação. Como o conjunto de termos é limitado, garante-se também a coerência do texto. (ESSELINK)

Reengenharia

Um problema enfrentado na Internacionalização de software é o espaço disponível para encaixar as traduções. O texto de orações traduzidas para outros idiomas (latino, neste caso), a partir do inglês, é em média 30% mais longo e pode chegar a 100% mais longo quando se trata de palavras soltas (ESSELINK).

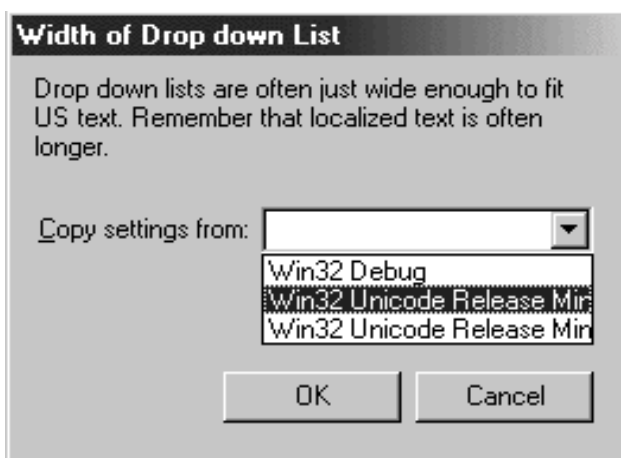


Figura 2 - Texto truncado

Pode-se tratar essa expansão de várias formas. Uma delas, usada com frequência, é simplesmente deixar um espaço a mais nas janelas e botões para comportar o texto que tem possibilidade de expansão.

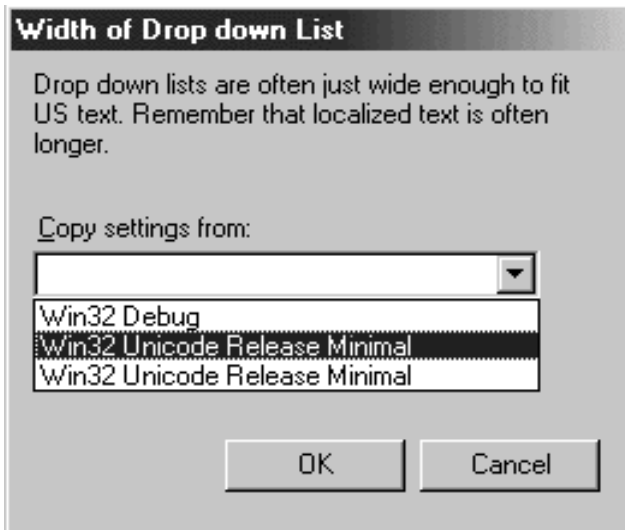


Figura 3 - Tamanho da janela expandido para comportar mais texto

Outra técnica consiste em utilizar layouts, disposição de elementos da interface, dinâmicos, que se ajustam automaticamente ao texto e a outros elementos inseridos, mas isso pode gerar uma situação em que o layout foge do controle do programador.

É necessário também que a interface do programa ofereça suporte para representação de diferentes caracteres. Quando do desenvolvimento, é preciso escolher um esquema de codificação que possa comportar todos os caracteres dos locais que o projeto deseja cobrir e que possibilite fácil expansão. Por exemplo, para suportar caracteres orientais, o esquema escolhido deve suportar o uso de caracteres double-byte (dois bytes). Outro exemplo é o caso específico da língua árabe, em que é preciso também suporte para escrita bidirecional. Para os desenvolvedores ocidentais, a melhor escolha é o Unicode, que suporta todas essas exigências e assim não aumenta a complexidade do desenvolvimento. Estas questões de codificação de caracteres são tratadas no item a seguir.

É preciso também escolher a linguagem de programação apropriada, de acordo com o ambiente operacional de desenvolvimento do software, com a experiência do programador, com o suporte existente para o conjunto de caracteres utilizado e também com o suporte da linguagem para o domínio da aplicação.

Representação interna de caracteres

Como já mencionado antes, línguas com uma quantidade consideravelmente grande de caracteres ou cuja representação difere da usada no Ocidente, onde se desenvolveu a maioria das tecnologias de informação, exigem outros sistemas de representação, conhecidos no setor como Esquemas de Codificação de caracteres.

Nos primórdios da informática, quando a tecnologia da informação se restringia aos Estados Unidos e a outros poucos países da Europa, e quase não havia intercâmbio de informações entre eles, cada qual trabalhou no seu próprio esquema de codificação, condizente com a língua que falava. Quando surgiu a Internet e ampliou-se o intercâmbio de informações entre os países, a tecnologia da informação passou a fazer parte do dia-a-dia de cada vez mais países. Neste momento, houve a necessidade de criar mais esquemas de codificações de caracteres para representar mais idiomas e estender os esquemas já existentes, também para comportar novas línguas. Estendeu-se o ASCII (“American Standard Code for Information Interchange” - Código Padrão Americano para Intercâmbio de Informações), usado nos Estados Unidos, para comportar caracteres latinos e assim representar português e espanhol, por exemplo.

Como todos sabemos, um computador entende apenas números, mais precisamente código binário, zeros e uns. O código ASCII, por exemplo, utiliza uma combinação de sete bits (um bit representa zero ou um) para representar todos os seus caracteres, num total de 127 caracteres possíveis, mais do que o necessário para comportar o inglês.

Mas qual a necessidade de usar um número fixo de bits para representar um caractere? Computação é uma ciência exata e determinística, que se utiliza muitas vezes de algoritmos (seqüências de passos pré-definidos) para realizar tarefas. Utilizando um número fixo de bits para representar um caractere, é possível dizer com certeza que a cada 7 bits sempre haverá uma letra, ou outro caractere qualquer, e isso facilita e muito o desenvolvimento de programas, pois é possível minimizar a verificação de erros.

Quando houve a necessidade de criar novos esquemas de codificação de caracteres, percebeu-se que somente 7 bits não seriam suficientes para representar todos os caracteres latinos, por exemplo. Era necessário utilizar uma quantidade maior de bits, de modo a poder comportar uma quantidade maior de caracteres. Era preciso também criar novos algoritmos de ordenação, que levassem em conta os novos caracteres. É aqui que entra o exemplo do “a” acentuado e do “a” craseado, citado na introdução deste artigo. O novo sistema que comportava todos os caracteres latinos tinha agora 8 bits e passou a se chamar ISO 8859-1⁴.

Mas, ainda que muitos esquemas de codificação estivessem normatizados pela ISO (“International Organization for Standardization” – Organização Internacional para a Padronização), existiam muitas incompatibilidades com sistemas próprios, em especial os criados no Oriente, que suportavam os conjuntos de caracteres coreanos, chineses complexos e simplificados, japoneses (diversos alfabetos), indonésios, etc. Era necessária uma codificação universal que permitisse representar vários tipos de texto, em línguas diferentes, dentro de um mesmo documento.

Para resolver esta situação, a ISO criou um novo padrão cujo propósito seria comportar o máximo possível de caracteres, a princípio usando dois bytes (16 bits) para representação de 65.536 caracteres diferentes. Esse sistema se chama ISO 10646 e a Unicode Consortium, organização privada sem fins lucrativos, criada por empresas e outros organismos interessados na padronização da representação de caracteres, levou-o adiante e tratou de fundamentá-

lo, padronizá-lo e estendê-lo ainda mais para permitir a representação de uma gama ainda maior de caracteres (HARRIS).

Hoje o Sistema Unicode (como é identificado agora) comporta mais de 96.447 caracteres diferentes. Existem codificações para o latim e já houve propostas de codificações até para Klingon (para os fãs de Jornada nas Estrelas) e Quenya (a famosa língua dos elfos da trilogia O Senhor dos Anéis).

Brincadeiras à parte, o Unicode existe com o propósito de armazenar informações de todas as línguas em meio digital, para que seja possível guardá-las com mais segurança para posterior estudo. Existem atualmente projetos acadêmicos que têm o intuito de codificar dentro do sistema Unicode vários alfabetos hoje pouco usados, línguas históricas como o Egípcio, por exemplo, e outras simbologias, como línguas de sinais, simbologia matemática, etc (UNICODE).

O Unicode Consortium desenvolveu vários esquemas de codificação compatíveis, inclusive com os esquemas ISO. O UTF-8 utiliza 8 bytes de representação e é compatível com o ISO 8859-1, o UTF-16 estende o UTF-8 mantendo o mesmo padrão e adicionando novas características e o UTF-32 é um esquema de codificação completo, que comporta todas as línguas já codificadas no mundo e possui ainda mais “espaço” para codificar muitas mais.

Os esquemas de codificação de caracteres tornaram-se tão importantes que os sistemas operacionais derivados do UNIX (Linux, BSD, MacOS) passaram a usar esquemas específicos para especializar-se ainda mais no locale ativo. Por exemplo, para especializar o locale pt-BR, é possível utilizar pt-BR.ISO8859-1 ou pt-BR.UTF-8.

Ferramentas de apoio à Internacionalização e à Localização

Conforme apresentado anteriormente, a internacionalização de um software permite que este possa lidar com as exigências de múltiplos locais, se necessário até ao mesmo tempo. Quanto à

interface visível pelo usuário e à documentação do software, a internacionalização prevê a separação completa entre os textos da interface e a lógica de funcionamento do programa, o que permite traduzi-los separadamente.

Há duas categorias principais de ferramentas de localização: as ferramentas lingüísticas (tecnologia lingüística) e as de administração e gestão, ou seja, as ferramentas concebidas para facilitar e automatizar fluxos de trabalho, processos, gerência de projetos, produtividade pessoal, etc. O setor da localização tem uma demanda considerável por ambas as categorias, afinal, a velocidade e o volume de trabalho envolvidos na localização, além da demanda de qualidade, tornam imperativo o uso de ferramentas. Além disso, muitos usuários têm conseguido economias consideráveis e outros benefícios com suas ferramentas.

A engenharia lingüística, a disciplina que produz tecnologia lingüística, é um setor relativamente novo e seu progresso depende da ascensão da computação e lida com uma das invenções mais complexas do planeta, a língua. Sendo assim, a tecnologia lingüística ainda não satisfaz às expectativas ingênuas de alguns de seus fundadores, que há muito tempo condenaram os tradutores humanos ao desaparecimento. Entretanto, surgiram diversos sistemas de tecnologia lingüística consagrados que auxiliam, em vez de substituir, os tradutores humanos. A interação construtiva entre homem e máquina parece destinada a durar por muito tempo.

As seções a seguir apresentam uma breve descrição das principais tecnologias envolvidas, que configuram o que se convencionou chamar de ferramentas CAT – “Computer Aided Translation” (Tradução assistida por computador):

Sistemas de gerenciamento terminológico

A terminologia, ou seja, os vocabulários especializados de setores e usos específicos, é a base em que se sustenta toda boa tradu-

ção. O uso correto da terminologia apropriada em sistemas de memória de tradução e de tradução automática (veja mais abaixo) melhora a qualidade e reduz o tempo posterior empregado em correção. Ao mesmo tempo, os tradutores humanos também necessitam de recursos terminológicos superiores e fáceis de usar. Isto é geralmente fornecido por sistemas de gerenciamento terminológico, bancos de dados de termos que contêm entradas em diferentes idiomas. Os bons sistemas são baseados em conceitos e não em palavras, permitem gerência efetiva de sinônimos e estão integrados a editores de texto para melhorar a produtividade do tradutor. O intercâmbio de terminologias entre diferentes sistemas de gerenciamento terminológico e entre estes e outras ferramentas, como sistemas de tradução automática, é objeto de um trabalho constante de padronização.

Memória de tradução (TM, do inglês “Translation Memory”)

As memórias de tradução são programas criados especialmente para tradução de textos longos e repetitivos, como manuais e documentos de ajuda de software. É um banco de dados que contém traduções realizadas anteriormente. Nele, os textos nos idiomas de origem e de destino foram previamente divididos em segmentos e alinhados uns aos outros. Quando se compara uma nova versão de um texto à versão antiga, o software de memória de tradução identifica os segmentos equivalentes traduzidos no texto antigo e insere-os no novo texto traduzido. Desta forma, o tradutor não precisa re-traduzir todas as frases novamente e tem garantida a coerência da tradução final. Além disso, a chamada “coincidência parcial” (em inglês *fuzzy matching*) permite que se insiram automaticamente os novos segmentos que são similares ao original, mas não idênticos, com a indicação de similaridade, para posterior edição por parte do tradutor, que avaliará quais são as alterações necessárias para que o novo texto fique corretamente traduzido.

Estas ferramentas, na maior parte das vezes, também exigem o tradutor de pensar na formatação do texto, pois “esconde” estes detalhes de forma que o tradutor possa concentrar-se no conteúdo e não na forma.

O uso eficiente de memória de tradução depende, dentre outras coisas, da qualidade da tradução original (pois os erros que ela contiver serão igualmente reproduzidos), do alinhamento (pois a reutilização poderá depender do tamanho dos segmentos identificados) e de como são tratadas as atualizações do texto de origem. Além disso, mesmo as “coincidências de 100%” (em inglês, 100% matches) poderão necessitar edição posterior para adequar-se a sutis alterações de contexto ou à adaptação dos textos ao mercado-alvo, como por exemplo, em um texto sobre automóveis, “freio” em português brasileiro e “travão” em português europeu.

Quando dos primórdios da tecnologia de memórias de tradução, cada programa, produzido por uma empresa diferente, usava seus próprios arquivos internos para manipulação de traduções, que apresentavam formatos internos diferentes. Mas, sempre houve o desejo de que fosse possível compartilhar informações entre tradutores usando ferramentas diferentes.

Pensando nisso, organizações internacionais de localizadores juntaram-se em projetos com o objetivo de criar padrões de arquivos para intercâmbio mundial de informações entre ferramentas de memórias de tradução. Entre os mais conhecidos e usados estão o TMX (“Translation Memory eXchange” – Intercâmbio de Memórias de Tradução) e o TBX (“TermBase eXchange” – Intercâmbio de Base de Dados de Termos), ambos concebidos e mantidos pelo grupo OSCAR (“Open Standards for Container/Content Allowing Re-use” – Padrões Abertos para Conteúdos que Permitam Reuso), da LISA, e o XLIFF, uma parceria do OSCAR com a Sun Microsystems.

A sintaxe e o formato desses arquivos encontram-se fora do escopo desse artigo. Mas, é bom saber que esses padrões foram concebidos para uso em escala mundial e que toda boa ferramenta de memória de tradução possui um conversor para um ou outro desses formatos.

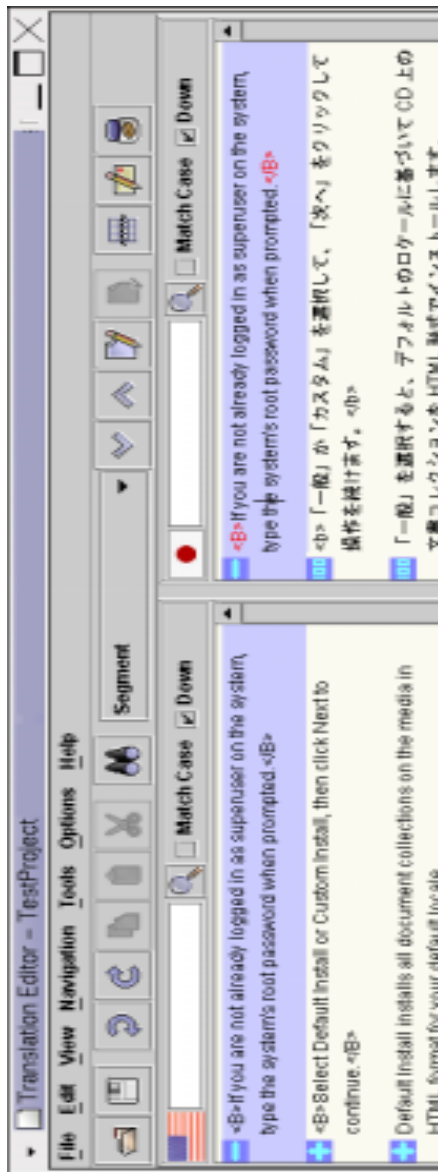


Figura 4 - Ferramenta de Memória de Tradução que usa XLIFF

Tradução automática (MT, do inglês “Machine translation”)

Esta tecnologia, muitas vezes mal compreendida, difere da memória de tradução porque efetivamente realiza análise lingüística nos textos em que é utilizada. Ao passo que as memórias de tradução são, em teoria, independentes do idioma, os sistemas de tradução automática dividem o texto de origem em seus elementos básicos antes de traduzi-los e remontá-los no idioma de destino. Os resultados não são comparáveis à tradução humana (de alta qualidade), mas podem oferecer ganhos de produtividade, mesmo que seja necessário editá-los posteriormente. Como alternativa, pode-se usar a MT sem edição posterior, para auxiliar na compreensão básica. Neste caso, isto é conhecido como tradução “meramente para informação” ou “gisting”⁵.

Uma das razões pelas quais a MT não consegue oferecer a mesma qualidade dos melhores tradutores humanos é porque as máquinas não conseguem trabalhar com a ambigüidade da mesma forma que as pessoas conseguem. Além disso, como as pessoas, os sistemas de MT têm problemas em compreender frases longas e rebuscadas. Como resultado, as traduções automáticas funcionam melhor em textos não-ambíguos, “técnicos” ou em áreas de assuntos restritos (“domínios”). Também são adequadas para grandes volumes de texto e produzem os melhores resultados quando integradas a outros processos de geração de documentos e de tradução. Entre outras coisas, isso ocorre porque a qualidade da redação e da edição aplicadas aos textos de origem é vital para seu sucesso.

A tradução automática depende muito da qualidade, do tamanho e da estrutura dos dicionários (conhecidos como léxicos) que o sistema utiliza. Por exemplo, traduzir um texto de TI somente com um léxico genérico não produzirá resultados de alta qualidade. Isto significa que pode ser necessário um tempo substancial de configuração para os sistemas de tradução automática, mas os resultados poderão ser muito bons, caso se identifiquem e acrescentem todos os termos pertinentes.

Outras ferramentas

Existe toda uma gama de ferramentas que podem ajudar um tradutor em qualquer parte do projeto da Localização de um software. As memórias de tradução e os programas de tradução automática são apenas algumas das ferramentas que podem ajudar no processo.

As demais ferramentas disponíveis auxiliam nos outros aspectos do processo de localização, como o gerenciamento do projeto, em outros aspectos específicos da localização da documentação, como a ajuda on-line, no desenvolvimento do website localizado do produto e no controle de qualidade. Várias das ferramentas estão disponíveis para avaliação no site do Localisation Research Centre (LRC).

Dentro do projeto de Localização de um software, o papel do tradutor extrapola as responsabilidades da tradução. Adquirida a experiência necessária, o profissional pode gerenciar projetos completos de localização e existem ferramentas criadas especialmente para auxiliar o profissional nesse processo.

Considerações Finais

Nesse artigo, abordou-se o assunto da Internacionalização e Localização de Software de forma introdutória, com os conceitos básicos do processo e dos atores envolvidos, apresentando ainda sistemas de codificação e ferramentas disponíveis para trabalhar nesse mercado. Procurou-se caracterizar que a ausência da internacionalização sistemática dos softwares acrescenta uma série de dificuldades à fase de localização, sendo, por isso, importante a adoção das técnicas de internacionalização para tornar a localização muito mais simples, mais eficiente e menos sujeita a falhas.

Enfatizou-se também a importância da localização e o fato de que a mesma não se restringe à tradução. Existem diversos outros processos envolvidos que podem ser avaliados mais profundamente na bibliografia associada. Também foi feita uma introdução a tipos de ferramentas que podem auxiliar o tradutor durante a localização e que permitem abrir um leque de possibilidades profissionais nesse mercado, bem como garantir um trabalho mais eficiente e preciso.

Entretanto, a área de tecnologia associada à Internacionalização e à Localização de software está, atualmente, em efervescência. O que era verdade e estável há um ano já não o é. As empresas e organismos internacionais que se dedicam a trabalhar e discutir a área representam uma comunidade crescente e em permanente movimento. É necessário estar atento às novas tecnologias e à evolução da padronização do setor (como os padrões XLIFF, TMX e Translation WS). Vemos, por fim, que a área é multidisciplinar por natureza.

O Brasil está entrando no mercado de exportação de software. Isso gerará muitas demandas e oportunidades de trabalho para profissionais da área de idiomas, mas estes devem estar preparados para esta realidade, com tecnologia e dinamismo. Há iniciativas como as do GeNESS [LISE⁶, OTIMILS⁷] para acompanhar e fornecer informações adequadas para os profissionais interessados neste novo mercado de trabalho.

Notas

1. É desnecessário dizer que a palavra Internacionalização é longa. Em inglês, a palavra Internationalization possui 20 letras. Alguém, que buscava uma forma reduzida de escrever esta palavra, teve a idéia de utilizar a seguinte lógica: utilizar a primeira e a última letra intercaladas pelo número de letras que ficam entre elas. Assim, criou-se a sigla I18N.
 2. A lógica de criação da sigla L10N é a mesma descrita para I18N.
 3. www.m-w.com
 4. O padrão ISO 8859 é composto por uma série de conjuntos de caracteres, identificados pelo número que segue o hífen final (ISO 8859-1, -2, -3, ...), representando diferentes subconjuntos do padrão completo, como o subconjunto de caracteres latinos, cirílico, grego, árabe, etc.
 5. O termo “gisting” também é utilizado em alguns casos para descrever o resumo de textos monolíngües, que não é o caso aqui.
 6. LISE – Simpósio de Localização e Internacionalização de Software para Exportação – <http://lise.geness.ufsc.br>
 7. OTIMILS – Observatório de Tecnologia e Inteligência de Mercado de Internacionalização e Localização de Software – <http://i18n.geness.ufsc.br>
- Obs.: As atividades de Internacionalização e Localização de software do GeNESS, aos quais os autores estão vinculados, tiveram o apoio da Finep em 2004.

Referências

ALBUQUERQUE, Alessandra. (2005). “Localização, Globalização, Localização e Tradução: Algumas considerações”. On line: em www.iscap.ipp.pt/cml/pgtac/ficheiros/teoria/Sistemas_TA.ppt. Acesso em 15/03/2005.

CADIEUX, Pierre; ESSELINK, Bert. (2002). “Globalization insider”. In: *GILT: Globalization, Internationalization, Localization, Translation*. v. 1, nº 5. On line: <http://www.lisa.org/globalizationinsider/>.

CID, Fabiano. (2004). “Multilingual Computing & Technology”. In *Localizing for Brazil*. Edição 62, Vol 14, nº 3.

CORRIGAN, John; FOSTER, Tim. *XLIFF: An Aid To Localization*. On line: developers.sun.com/dev/gadc/technicalpublications/articles/xliff.html

DePALMA, Don. (2002). *Business Without Borders: A Strategic Guide to Global Marketing*. New York: John Wiley & Sons; 1a. edição.

ESSELINK, Bert. (2000). *A Practical Guide to Localization*. John Benjamins Publishing. Development Editor: Arjen-Sjoerd de Vries. Copy Editor: Shiera O'Brien.

HARRIS, John; MCCORMACK, Ryan. (2000). “Translation is not enough. Considerations for global Internet development”. On line: www.sapient.com/pdfs/strategic_viewpoints/globalization_a4.pdf. Acesso em 15/03/2005.

LOMMEL, Arle & FRY, Debora. (2005). Manual de Introdução à Localização. Título original: The Localization Industry Primer. 2a. edição. On line: www.lisa.org/products/primers/primer2_poBR.pdf. Acesso em 15/03/2005.

LOCALIZATION INDUSTRY STANDARDS ASSOCIATION. On line: www.lisa.org. Acesso em 15/03/2005.

LRC (Localisation Research Centre). Industry Software Localisation Tools. On line: <http://www.localisation.ie/exchange/loctools.htm>. Acesso em 15/03/2005.

UNICODE Consortium. (2004). The Unicode Standard Version 4.0.1. On line: www.unicode.org/versions/Unicode4.0.1. Acesso em 15/03/2005.