

# 共同ローカリゼーションフレームワーク —ソフトウェアローカリゼーション(翻訳)を支援する Web アプリケーション—

## 1. 背景

近年、オープンソースソフトウェアが広く一般にも普及してきた。

これらのソフトウェアには、商用ソフトウェアと比較しても遜色のない品質や機能を有したものが多数ある。しかしながら、優秀なオープンソースソフトウェアの多くは世界中の開発者によって連携して作られており、ユーザインタフェースやドキュメントが英語で書かれている方が一般的である。

当該ソフトウェアの利用者がソフトウェア技術者のみであれば英語であっても問題は少ないであろうが、エンドユーザにとっては母語でないと活用は難しい。以前に比べて、日本語での利用に対応したものも増えているが、ユーザ数が多くないソフトウェアなどでは、いまだ日本語化されていないソフトウェアも数多く存在しているのが現状である。

そこで、このようなオープンソースソフトウェアをより 広く一般のユーザにも普及させるために、ユーザインタフェースやドキュメントの日本語化が必要となる。

## 2. 目的

ソフトウェアの各言語対応は図 1 に示すように 2 つのプロセスに分けられる。一つはインターナショナル化(国際化)といって開発者サイドでソフトウェア自体を翻訳可能にする作業であり、もう一つはローカリゼーション(地域化)といって翻訳者サイドで実際に翻訳を行う作業である。

インターナショナル化では、GNU gettext のような標準的なライブラリやツールが整備されており、多くのオープンソースプロジェクトが多言語対応を意識して開発を行っている。

ローカリゼーションでは、例えば GNU gettext が生成する PO ファイルに対応したエディタであったり、Mozilla 製品の拡張機能(アドオン)に対応した Web アプリケーションといったツール類が存在するものの、汎用的な支援ツール・エディタなどは少ない。また、翻訳業界では翻訳メモリ(TM=Translation Memory)というデータベースを使って、一度翻訳した内容をコンピュータで記憶しておいて翻訳作業を効率化する機能が幅広く使われているが、ソフトウェアローカリゼーションで利用されるようなソフトウェアでは翻訳メモリに対応したものは非常に少ない。

そこで、本プロジェクトでは、ソフトウェアローカリゼーションという枠組みにおいて、翻訳メモリなどの技術を用いた汎用的なローカリゼーション支援ソフトウェアを開発する。特にオープンソースソフトウェアといったチームで翻訳作業を行うことを想定し、データや情報を共有することで最大限の効率化を図る。

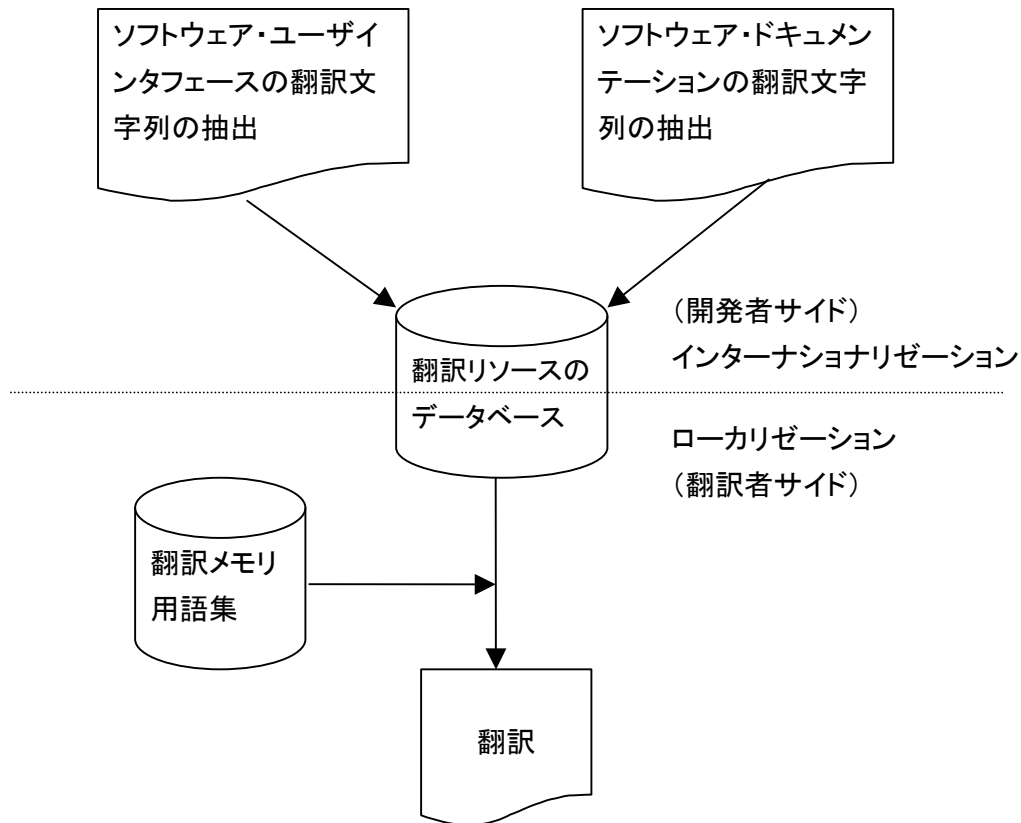


図 1 ソフトウェアの翻訳プロセス

### 3. 開発の内容

本プロジェクトでは、共同ローカリゼーションフレームワーク LxL(発音は el-eks-el:エレクセル)を開発した。LxL の名前は、数学の「直積集合」を求める記法に由来し、ローカリゼーション作業が、ある言語とある言語のペア(言語対)を作り出すことよりネーミングした。

#### 3. 1. システムの構成

LxL は、Web アプリケーションとして実装した。サーバ側では、オープンソースソフトウェア環境で広く利用されている LAMP を、また、クライアント側では Ajax が動作する環境を前提とした。LxL の標準的な動作環境を表 1 にまとめる。

表 1 LxL の動作環境

サーバ	クライアント
LAMP 環境を想定: <ul style="list-style-type: none"> <li>• Linux</li> <li>• Apache</li> <li>• MySQL</li> <li>• Perl 5.8 以降</li> </ul>	Ajax が正常動作する環境 具体的には以下を想定: <ul style="list-style-type: none"> <li>• IE 6.0 以降</li> <li>• Firefox 2.0 以降</li> <li>• Safari 2.0 以降</li> <li>• Opera 9 以降</li> </ul>

現在の Web アプリケーション開発では Web フレームワークを使用するのが主流である。しかし、インストールが簡単に行え、コンパイルが不要な Pure Perl モジュールのみに依存

した優れた Web フレームワークが存在しないことから、汎用の Web フレームワークを自作した。

ユーザの操作性に配慮し、クライアント側では Ajax インタフェースを導入した。具体的には、翻訳文を入力するリッチテキストエディタ部並びに翻訳メモリより類似文の検索結果を候補として提示するオートコンプリート機能のプロトタイプを開発した。

### 3. 2. 翻訳メモリ

翻訳支援機能として翻訳メモリを実装した。翻訳メモリとは、原文と翻訳文のペアをデータベース化したもので、一度翻訳された情報を再利用することで、翻訳作業の効率化と質の向上を支援する。

通常、翻訳メモリは各翻訳者や各プロジェクト内の翻訳結果のみを集めて作られるが、LxL ではプロジェクトをまたいで翻訳メモリを自動的に広く共有可能である。このため、他者の翻訳結果を簡単に活用することで、翻訳作業の効率化と翻訳文のさらなる質の向上が期待できる。

翻訳メモリ内に発見された過去の訳文が表示されている例を図 2 に示す。なお、右側のテキストボックスに翻訳文を入力することでローカライズが可能になっている。

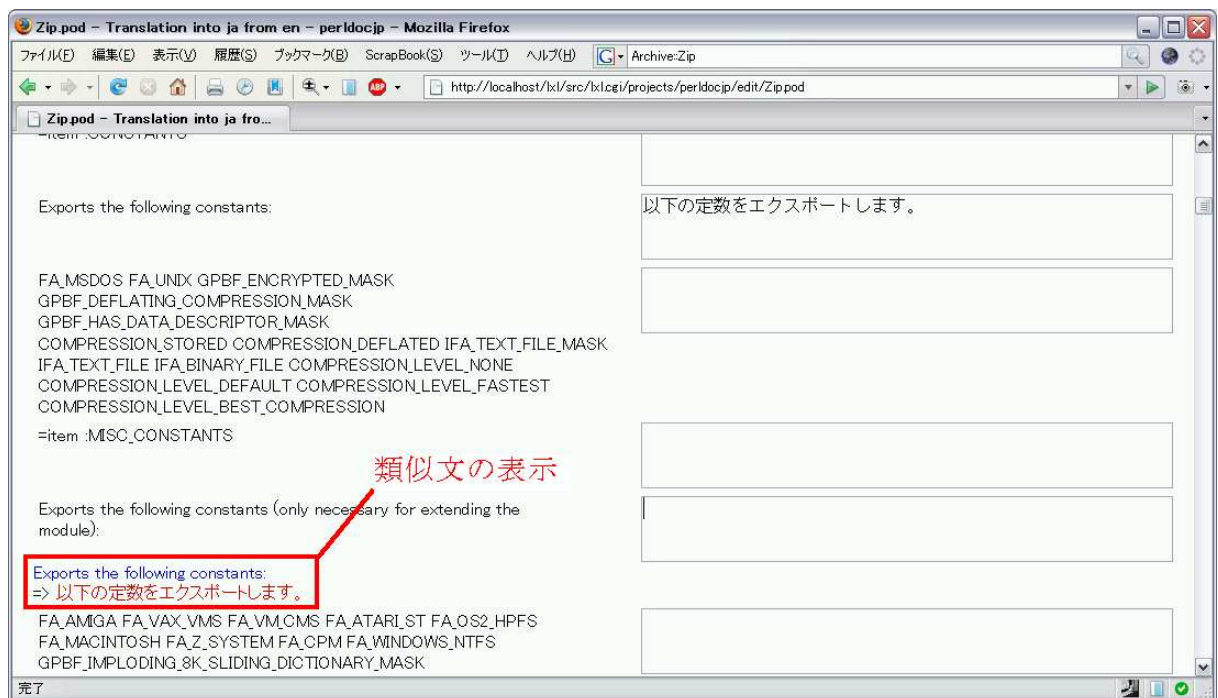


図 2 LxL の編集画面 (翻訳メモリからの類似文の表示)

### 3. 3. チーム作業支援

チーム作業の支援機能として、ユーザ管理機能、プロジェクト管理機能、権限管理機能を実装した。

プロジェクト管理機能としては、各ファイル単位で翻訳の進捗状況を確認する機能などがある。また、権限管理機能として、各ファイルに対してローカライズ担当者を割り当てるといった操作が可能である。

#### 4. 従来の技術(または機能)との相違

本プロジェクトで開発した LxL はローカリゼーション支援アプリケーションであるが、その機能面の特徴としては「Web アプリケーション」であること、役割分担などの「チーム作業を支援」すること、「翻訳メモリを搭載し作業を効率化」できることの 3 点に集約される。

ローカリゼーション支援アプリケーション自体は 既に多く存在している。しかし、既存の「チーム作業を支援」する「Web アプリケーション」は「翻訳メモリを搭載」していない。また、既存の「翻訳メモリを搭載し作業を効率化」に対応できているものは「デスクトップアプリケーション」であり、「チーム作業の支援」に限界がある。

「チーム作業の支援」と「翻訳メモリを搭載し作業を効率化」の両者に対応しているものが存在しない中で、その両者を満たして設計されている LxL はユニークであり、有用である。現状では、本アプリケーションは開発途上でありプロトタイプレベルを抜けておらず、各々の機能では既存のアプリケーションの 域に達していない点もあるが、今後開発を進め、機能が成熟してきた際には期待が高いものである。

#### 5. 期待される効果

本ソフトウェアにより、ソフトウェアローカリゼーションの効率化が期待できる。特に、オープンソースソフトウェアにおいては、翻訳メモリを共有することで、より早く・より多く・より正確な翻訳を提供することが容易になる。その結果、エンドユーザにとってオープンソースソフトウェアがさらに使いやすいものとなり、ユーザの増加が見込まれる。

普段からオープンソースソフトウェアに慣れ親しむユーザが増えることで、オープンソースコミュニティや開発に参加するユーザも増え、ソフトウェア産業の発展に寄与することに繋がる。さらに、本ソフトウェアは英語のソフトウェアを日本語にローカライズするのみならず、日本語のソフトウェアを英語や他の言語にローカライズするためにも活用でき、日本発のオープンソースソフトウェアを海外に広めるためにも有用である。

#### 6. 普及(または活用)の見通し

本ソフトウェアの主なターゲットユーザは、オープンソースソフトウェアの開発者とそれに関わる翻訳者である。まずは筆者の関わるプロジェクトで利用し、その有用性を実証する。オープンソースソフトウェア関連のコミュニティは横の繋がりが大きいため、実際に使えることが分かれば徐々に世界中で広まっていくことを期待する。

#### 7. 開発者名(所属)

井上 謙次 (大阪工業大学 大学院 情報科学研究科 博士前期課程)