

오픈소스 소프트웨어 한글화방법에 관한 연구

이윤호*, 김광명**, 윤희용**

*성균관대학교 컴퓨터공학과

**성균관대학교 정보통신공학부

e-mail: {leeyunho83, keikim}@skku.edu, youn@ece.skku.ac.kr

Study on Open Source Software Localization

Yun Ho Lee*, Kwang Myung Kim**, Hee Yong Youn**

*Dept. of Computer Engineering, Sungkyunkwan University

**School of Information and Communication Engineering, Sungkyunkwan University

요 약

요즈음 리눅스 및 안드로이드 운영체제를 필두로 오픈소스 소프트웨어 사용이 증가하고 있고 언어의 다양성, 전통 및 문화가 프로그래밍 세계에 반영되어야 할 필요가 있다. 본 논문은 기존에 이루어졌던 특정 소프트웨어 위주의 주먹구구식 로컬라이제이션에서 벗어나서 체계적인 로컬라이제이션 프로세스 모델을 확립해서 소프트웨어의 활용성을 높일 뿐 아니라 국내 소프트웨어의 글로벌화적인 면에서 고려가 될 수 있다.

1. 서론

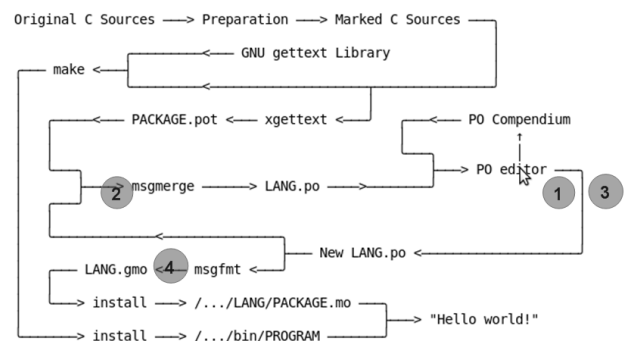
오픈소스 소프트웨어는 이미 정보 기술 분야에서 주류에 해당한다. 세계 유수의 기업에서는 사용방식에서의 차이만 있을 뿐 오픈소스를 사용하지 않는 곳은 없다. 비용 문제를 떠나서 오픈소스 소프트웨어는 상용경쟁제품에 비해 유연함을 갖추고 있는 것으로 간주된다. 오픈소스를 대표하는 리눅스를 한글화하는 작업이 있었고 그로 인해 영어에 익숙하지 않은 일반사용자에게 편리함을 가져다주었다. 요즈음 안드로이드 운영체제로 대표되는 오픈소스 소프트웨어사용의 증가로 인해 한글화에 관한 관심 또한 증가하고 있다. 본 논문은 일반적인 로컬라이제이션을 좀 더 실용적으로 적용할 수 있는 그 절차를 규명하고 정의하는데 그 일차적인 목적을 두며 리눅스 기반의 영문문화권 소프트웨어의 한글화를 위한 프로세스 모델을 정의한다.

1.1 기존 한글화 방법의 문제점

특정 한글화 커뮤니티를 기준으로 일반 텍스트 파일을 열어서 제대로 된 번역도구의 사용 없이 번역하는 일이 많았으며, 리눅스 배포판의 일종인 우분투의 한글화 방법을 예로 들어 살펴보면 런치패드라는 사이트에 접속하여 자신이 관심 있는 부분의 번역을 해서 올리고 한국어 번역 팀의 코디네이터가 번역내용을 확인하고 Accept하는 방식이다. 하지만 이 방법은 기존에 번역된 번역물과 병합 없이 수많은 사용자들이 새롭게 번역하는 경우가 많았기 때문에 번역의 질과 생산성을 높이기 위해서는 기존 번역을 토대로 수정해야 하는 부분만 수정한 다음 기존 번역

물과의 병합을 통해 생산성을 높일 수 있는 번역방법이 필요하다. 본 논문에서는 poedit이라는 po파일 전문 번역도구를 사용하여 번역하고 msgmerge라는 프로그램을 이용하여 기존 번역물과의 병합을 통해 번역물의 재활용성을 높이는 방법에 대하여 알아본다.

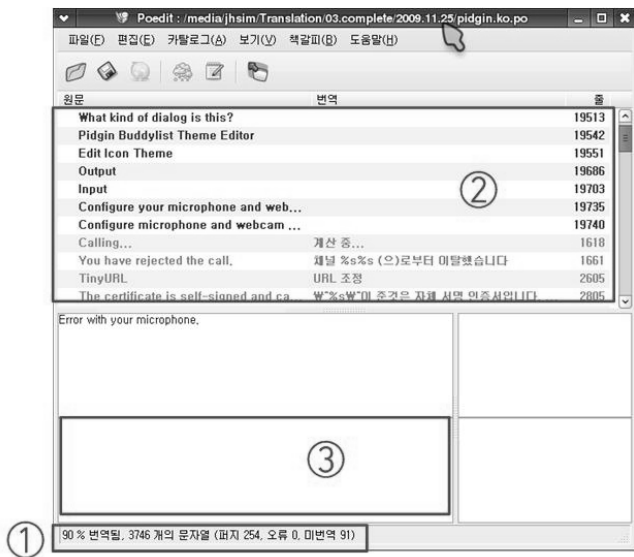
2. 대표적인 OSS인 리눅스의 한글화



(그림 1) 번역 절차

1. po 파일번역: poedit, virtaal같은 번역도구를 이용하여 메시지를 번역한다.
2. merge 작업: 앞서 번역한 파일과 병합한다.
3. 재번역 및 검토: 병합한 파일에서 잘못된 부분을 찾아 수정한다.

2.1 메시지 번역



(그림 2) 리눅스 번역 과정

1. 번역상태를 확인한다.
2. 번역할 메시지를 선택한다.
3. 메시지를 번역한다.

2.1.1 번역물 병합

```
msgmerge def.po ref.po(t) -o result.po
```

msgmerge 기존 번역 새로운 번역 또는 서식 -o 결과
msgmerge : 두 개의 동일한 po파일을 하나로 병합한다.
옵션: -o, --output-file=<파일> 출력 파일을 지정한다.

2.1.2 바이너리 추출

po는 원문을 참조하여 사람이 이해할 수 있는 언어로 번역한 파일입니다. 따라서 시스템에 번역을 적용하려면 기계가 인식할 수 있는 기계어 즉, 바이너리를 추출해야한다.

```
msgfmt translation.po -o translation.mo
```

msgmerge 번역파일(.po) -o 바이너리(.mo)

mo 파일의 이름은 /usr/share/locale/ko/LC_MESSAGES 디렉터리의 해당 파일 이름이 기준이 된다.

msgfmt : po파일에서 바이너리파일인 .mo파일을 생성한다.
옵션: -o, --output-file=<파일> 출력 파일을 지정한다.

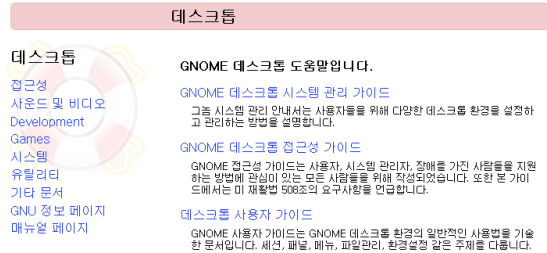
번역 내용을 확인하기 위해 .mo 파일을 복사해서 붙여넣는다.

```
# cp translation.mo
```

```
/usr/share/locale/ko/LC_MESSAGES/translation.mo
```

프로그램을 띄워서 번역 결과를 확인한다.

2.2 도움말 번역



(그림 3) 리눅스 도움말 화면

2.2.1 xml 추출

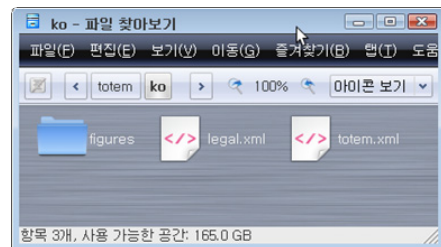
한글 도움말 디렉터리가 없는 경우에는 /usr/share/gnome/help/프로그램이름 경로에서 en_GB라는 이름의 디렉터를 복사해서 이름을 ko로 변경한다.

2.2.2 omf 파일 추출

omf 파일이 없어서 번역된 도움말이 보이지 않을 수 있다. /usr/share/omf/프로그램이름 경로에서 *.en_GB.omf 파일을 복사하여 *.ko.omf로 변경한다.

2.2.3 도움말 구성

도움말 디렉터리는 프로그램명.xml, legal.xml과 사용한 이미지를 담고있는 figures 디렉터리로 구성된다.



(그림 4) 도움말 디렉터리 구성

프로그램명.xml : 실제 도움말 본문 내용을 다룬다.

legal.xml : 문서의 복제 및 재배포에 관한 내용을 다룬다.

figures : 도움말에 사용되는 이미지이며 확장명은

.png이다.

2.2.4 xml 파일 편집

```
<inlinemediaobject>
  <imageobject>
    <imagedata
      fileref="figures/totem_pause_button.png" format="PNG"/>
    </imageobject>
    <textobject>
      <phrase>Shows pause button.</phrase>
    </textobject>
  </inlinemediaobject>
```

결과: totem_pause_button.png 이미지가 삽입된다.

```
<menuchoice>
  <guisubmenu>프로그램</guisubmenu>
  <guimenuitem>그래픽</guimenuitem>
  <guisubmenu>이미지 뷰어</guisubmenu>
</menuchoice>
```

결과: 프로그램 ▶ 그래픽 ▶ 이미지뷰어

```
<orderedlist>
  <listitem>
    <para>
      <guilabel>프로세스</guilabel> 탭을 선택하여 프
      로세스 목록을 표시한다.
    </para>
  </listitem>
  <listitem>
    <para>
      <menuchoice><guimenu>보기
      </guimenu><guimenuitem>의존성
      </guimenuitem></menuchoice>을 선택한다.
    </para>
  </listitem>
</orderedlist>
```

결과:

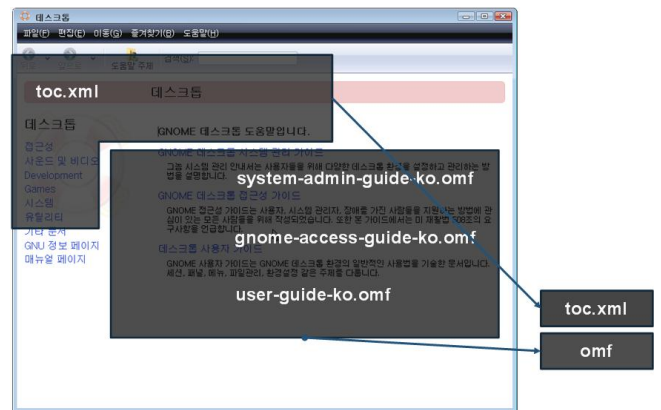
1. 프로세스 탭을 클릭하여 프로세스 목록을 표시한다.
2. 보기 ▶ 의존성을 선택한다.

2.2.5 omf 파일 편집

아래의 코드에서처럼 한국어 언어코드인 ko로 바꿔준다.

```
<?xml version="1.0" encoding="utf-8"?>
<omf>
  <resource>
    <creator>(Sun GNOME Documentation Team)</creator>
    ...
```

```
<maintainer>(GNOME Documentation Project)</maintainer>
<title>데스크톱 사용자 가이드</title>
<date>May 2002</date>
<version identifier="GNOME 2.0 Desktop User Guide V1.0"
date="May 2002"/>
<subject category="GNOME|Desktop"/>
<description>
GNOME 사용자 가이드는 GNOME 데스크톱 환경의 일반적인 사용법을
기술한 문서이다.
세션, 패널, 메뉴, 파일관리, 환경설정 같은 주제를 다룬다.
</description>
<type>user's guide</type>
<format mime="text/xml" dtd="//OASIS/DTD DocBook XML
V4.3//EN"/>
<identifier
url="file:///usr/share/gnome/help/user-guide/ko/user-guide.xml"/>
<language code="ko"/>
<relation seriesid="6a60980a-7315-11d6-9cf7-fdf5569be9f8"/>
<rights type="GNU FDL" license.version="1.1" holder="Sun
Microsystems"/>
</resource>
</omf>
```



(그림 5) 각 파일의 적용범위

gnome 데스크톱 시스템 관리가이드, gnome 데스크톱 접근성 가이드, 데스크톱 사용자 가이드는 각각 다른 omf파일의 내용입니다.

3. 기대효과

본 연구를 통해 기존에 특정 어플이나 커뮤니티 위주의 주먹구구식으로 이루어졌던 로컬라이제이션에서 벗어나서 체계적인 로컬라이제이션 프로세스 모델을 확립해서 선진 소프트웨어의 활용성을 높일 뿐 아니라 국내 소프트웨어의 글로벌화적인 면에서 고려가 될 수 있다.

4 결론

대표적인 오픈소스 소프트웨어인 리눅스의 한글화 과정에 대하여 살펴보았다. 이미 국내에는 리눅스 한글화 프로젝트, gimp 한글화 프로젝트 등 수많은 오픈소스 한글화 작업이 진행 중에 있다. 또한 다양한 종류의 안드로이드폰이 출시되면서 외국 어플을 한글화하는

커뮤니티도 많이 생겨나는 추세이다. gnu gettext를 이용하는 측면에서 번역방법은 위에 소개한 내용과 크게 다르지 않을 것이나 번역의 질 측면에서는 번역자에 의존할 수밖에 없는 현실이다. 앞으로 공통된 번역어사전도 만들어져 제대로 한글화된 오픈소스 소프트웨어가 많아지길 기대해본다.

참고문헌

- [1] 안선희, 김해진, 남궁한, 손덕주. "UNIX 운영체제에서의 국제화와 한글화 방향에 관한 연구.", 한국정보과학회 언어공학연구회, 44-49. 1989.
- [2] 이장호, 조유근. "UNIX 한글화 구현방안의 비교." 한국정보과학회, 537-540. 1990
- [3] 신승근, "한글화 로컬라이제이션 프로세스 모델 사례 연구", 숭실대학교, 2003
- [4] Fernando Bento. Open source ERP's I18n. In Proceedings of the Workshop on Open Source and Design of Communication (OSDOC '10). ACM, New York, NY, USA, 49-50. 2010.
- [5] Witte, Carsten. Improved Usability through Internationalization. Springer Berlin / Heidelberg. 2011
- [6] Steffen Gross, "Internationalization and Localization of Software", Eastern Michigan University Department of Computer Science