NGHIÊN CỨU SỬ DỤNG GETTEXT ĐỂ ĐA NGỮ HOÁ PHẦN MỀM

A STUDY ON USING GETTEXT FOR MULTILINGUALIZING SOFTWARE

VÕ TRUNG HÙNG
Trường Đại học Bách Khoa, Đại học Đà Nẵng
ĐẶNG QUỐC VIỆN
Trung tâm ứng dụng công nghệ thông tin và truyền thông – DATIC

TÓM TẮT
Trong bài báo này, chúng tôi giới thiệu những kinh nghiệm của chúng tôi khi đa ngữ hoá các phần mềm bằng cách sử dụng GETTEXT. Chúng tôi trình bày những vấn đề tổng quát nhất khi sử dụng GETTEXT như cách cài đặt, cách tổ chức quản lý các thông điệp đa ngữ, trình tự thực hiện khi đa ngữ hoá một chương trình và cách khai thác các hàm.

ABSTRACT
In this paper, we present our experience in the multiligualization of a software by using GETTEXT. We introduce importants points of GETTEXT such as the installation, organization and management of multilingual messages, process of the multiligualisation a programme and using the functions.

1. Giới thiệu

Sự sử dụng phổ biến của Internet đã tạo ra một môi trường toàn cầu hóa về thông tin, kèm theo nó là nhu cầu đa ngữ hóa các phần mềm, trang Web. Vấn đề trao đổi thông tin giữa các dân tộc, giữa các cộng đồng trên thế giới với nhau luôn gặp phải khó khăn và một trong những khó khăn lớn nhất là sự khác biệt về ngôn ngữ.

Trước đây, các phần mềm, trang Web thường được viết trong tiếng Anh nhưng thực tế có rất nhiều người trên thế giới không thể đọc và hiểu được tiếng Anh (theo thống kê của World Information Access Project thì Internet được truy cập ở 165 quốc gia và có 69% người sử dụng Internet hoàn toàn không sử dụng được tiếng Anh). Có nhiều người tuy sử dụng được tiếng Anh nhưng họ vẫn yêu thích làm việc trên những phần mềm, website trong ngôn ngữ mẹ đẻ của mình.

Các nhà sản xuất phần mềm luôn mong muốn bán được nhiều sản phẩm không những trong nước mà còn ở nước ngoài và một trong những yêu cầu để bán được sản phẩm ở nước ngoài là phần mềm của họ phải có khả năng đa ngữ (cho phép người sử dụng chọn lựa ngôn ngữ khi làm việc). Vì vậy, nhu cầu đa ngữ hoá phần mềm đặt ra một cách tự nhiên. Cùng với quá trình toàn cầu hoá, việc đa ngữ hoá và bản địa hoá phần mềm đang là vấn đề được các nhà khoa học, các nhà sản xuất phần mềm quan tâm.

Khi phát triển các ứng dụng cho người sử dụng bản địa, các chuyên gia phần mềm đứng trước thách thức làm sao cho tiếng bản địa trong các ứng dụng phải thể hiện đúng và đầy đủ bản sắc riêng của từng ngôn ngữ chứ không đơn thuần là việc dịch văn bản. Hơn nữa, ngoài các chức năng chung của phần mềm người ta còn quan tâm đến các chức năng khác mang tính đặc thù của ngôn ngữ như hỗ trợ mã hóa, bộ gõ, phông chữ... Bên cạnh vấn đề hiển thị được các ký hiệu bản địa đặc trưng, người sử dụng bản địa còn muốn các ứng dụng trên máy tính

của họ đáp ứng được những tập quán, quy ước của ngôn ngữ viết, định dạng về ngày tháng, tiền tệ, thứ tự sắp xếp...

Trong bài báo này, chúng tôi tập trung giới thiệu các mô hình quản lý các thông điệp (message) đa ngữ và công cụ gettext để hỗ trợ phát triển các phần mềm đa ngữ.

2. Tổng quan về đa ngữ hoá phần mềm

2.1 Khái niệm

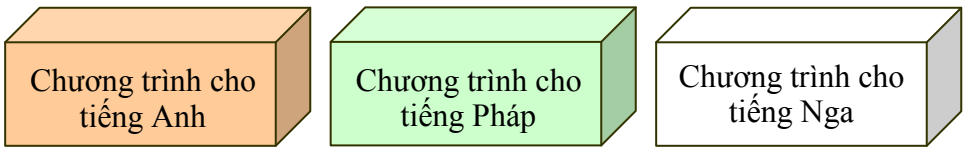
Phần mềm đa ngữ (multilingual software) là phần mềm cho phép người sử dụng chọn lựa ngôn ngữ (ví dụ: tiếng Anh, tiếng Pháp, tiếng Việt, ...) khi làm việc với phần mềm đó. Ví dụ: Windows cho phép người sử dụng chọn lựa 1 trong 43 ngôn ngữ khác nhau.

Ngoài ra, nó cung cấp các công cụ hỗ trợ người dùng làm việc trên các ngôn ngữ khác nhau. Những hỗ trợ này có thể là: hỗ trợ bộ mã, bộ gõ, hiển thị, xử lý ngôn ngữ như sắp xếp, dò lỗi...

2.2 Các mô hình tổ chức quản lý các thông điệp

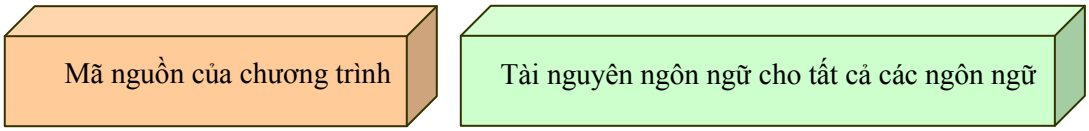
Có 3 mô hình tổ chức phần mềm đa ngữ thường được sử dụng hiện nay là:

– **Mô hình 1:** Phương pháp truyền thống là các thông điệp được viết gắn liền trong mã nguồn chương trình. Với mô hình này, người ta viết chương trình trong một ngôn ngữ xác định (ví dụ tiếng Anh) và sau đó tạo ra một bản sao rồi sửa lại cho thứ tiếng khác (ví dụ cho tiếng Pháp)… Với kiểu tổ chức này, mỗi khi sửa đổi mã nguồn ta phải tiến hành sửa đổi trên tất cả các tập tin chương trình tương ứng với các ngôn ngữ khác nhau.



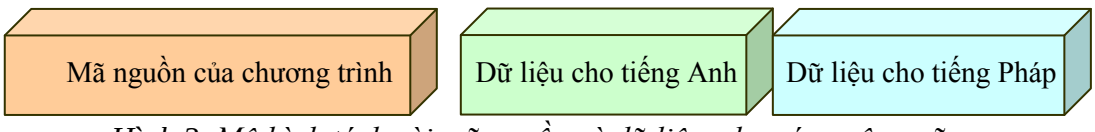
Hình 1. Mô hình tổ chức phần mềm đa ngữ bằng phương pháp truyền thống

– **Mô hình 2:** Phương pháp tách rời mã nguồn chương trình với các dữ liệu ngôn ngữ. Các dữ liệu ngôn ngữ (thông điệp, các hàm…) đặt trong một tập tin dùng chung cho tất cả các ngôn ngữ.



Hình 2. Mô hình sử dụng tài nguyên chung cho tất cả các ngôn ngữ

– **Mô hình 3:** sử dụng phương pháp tách rời mã nguồn với các dữ liệu ngôn ngữ và dữ liệu của các ngôn ngữ là lưu trữ riêng biệt cho mỗi ngôn ngữ.



Hình 3. Mô hình tách rời mã nguồn và dữ liệu cho các ngôn ngữ

2.3 Một số công cụ hỗ trợ đa ngữ hoá

Hiện tại có những hệ thống như Catgets, Ariane-G5, Gettext cho phép "bản địa hóa" (localisation) dễ dàng những tập tin thông điệp nhưng chúng chưa thể xử lý những biến đổi ngôn ngữ bên trong những thông điệp [7].

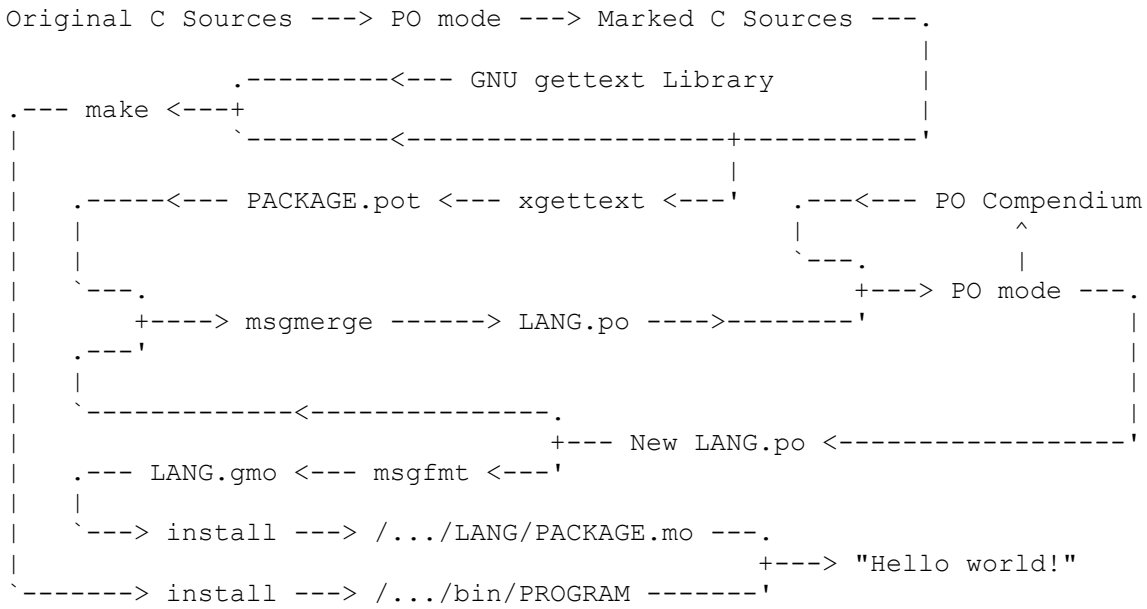
3. Gettext

3.1 Giới thiệu

Gettext là một phần mềm mã nguồn mở phục vụ bản địa hóa phần mềm và được dùng cho một số ngôn ngữ thông dụng. Gettext hỗ trợ dịch những thông điệp từ một ngôn ngữ nào đó ra ngôn ngữ mà người sử dụng có thể đọc được ngay. Hiện tại Gettext hỗ trợ bản địa hóa cho 12 ngôn ngữ khác nhau và mỗi ngôn ngữ sử dụng một hệ thống mã hóa khác nhau hay có thể chuyển mã về một hệ thống mã duy nhất là UTF-8.

3.2 Qui trình xử lý trong Gettext

Để bản địa hóa một chương trình, quá trình thực hiện trong gettext như sau:



Hình 4. Qui trình bản địa hóa một chương trình bằng gettext

Đối với việc bản địa hoá phần mềm, chúng ta cần phải dùng một số tiện ích giao diện dòng lệnh là xgettext, msgmerge, msgfmt, gettext, ngoài ra để soạn thảo các file PO(T) có thể dùng một công cụ giao diện đồ hoạ chạy trong KDE là KBABEL.

3.3 Cài đặt

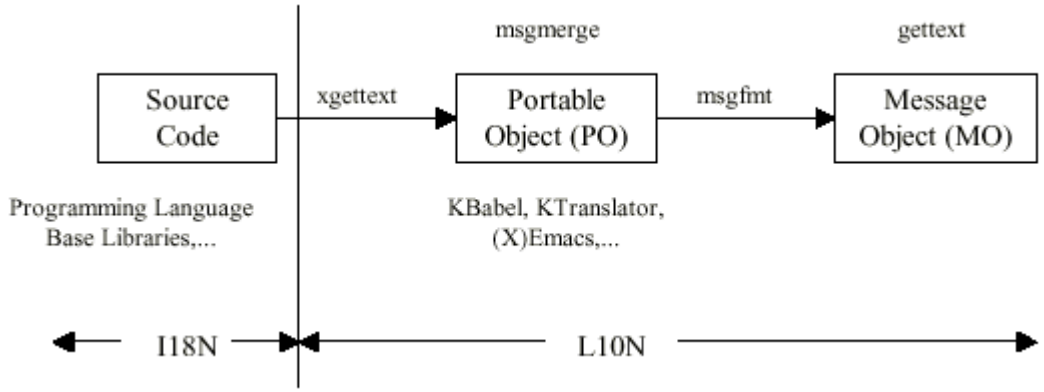
Bộ công cụ gettext được sử dụng rộng rãi để bản địa hóa phần mềm. Chúng ta có thể kiểm tra xem đang có phiên bản gettext nào đã cài đặt trên máy (dùng hệ điều hành Unix hoặc Linux) bằng cách gõ vào gettext: --v, hiện tại chúng ta có thể tải phiên bản gettext 0.16.5 tại địa chỉ *http://ftp.gnu.org/gnu/gettext*. Để cài đặt và cấu hình Gettext, chúng ta có thể tham khảo trên trang Web *http://www.gnu.org/software/gettext/manual/gettext.html*.

Sau khi có bộ cài đặt, ta tiến hành cài đặt như sau:

```
./configure --prefix=/usr &&
make &&
make install
```

3.4 Trình tự thực hiện

Để tiến hành đa ngữ hoá một chương trình theo gettext ta phải thực hiện các 4 bước sau:



Hình 5. Trình tự thực hiện địa phương hoá một ứng dụng

Bước 1: Sửa mã nguồn chương trình theo qui định của gettext

Ta cần chỉnh sửa mã nguồn theo `gettext` ở dạng I18n (Internationalization) và tạo môi trường làm việc (*setlocale*) theo ngôn ngữ cần bản địa hóa.

Trong chương trình, ta cần phải thay các thông điệp bằng cách gọi thực hiện hàm gettext. Ví dụ : thay vì viết

```
printf (“Hello world !”) ;
```

thì ta phải viết lại :

```
printf (gettext(“Hello world !”)) ;
```

Bước 2: trích các thông điệp ra tập tin riêng.

Sau khi sửa mã nguồn, ta thực hiện chương trình xgettext để trích các thông điệp (tham số của hàm gettext) đưa vào tập tin thông điệp có cấu trúc như sau:

```
#: <Tên chương trình>:<Dòng hướng dẫn của chương trình>
msgid "Thông điệp nguồn"
msgstr ""
```

Tập tin này do gettext tự động tạo ra (có phần mở rộng là PO - Portable Objects) và chứa các thông điệp gốc ***msgid*** và các thông điệp sau khi dịch ***msgstr***.

Bước 3: Dịch các thông điệp ra ngôn ngữ bản địa

Ta phải dịch các thông điệp gốc sang ngôn ngữ cần bản địa hóa. Ví dụ:

```
msgid "Hello world !"
msgstr "Chào mọi người"
```

Ngoài ra, chúng ta có thể dùng các công cụ của gettext để xử lý các tập tin thông điệp như kết hợp hai tập tin (***msgcat***), dịch lại tập tin và khớp thông tin chính xác cho thông điệp ***msgmerge*** , trích tự động các thông điệp ***xgettext*** (`extract gettext`),. . .

Khi cần tìm hiểu sâu về vấn đề quốc tế hoá và bản địa hoá phần mềm, chúng ta có thể sử dụng công cụ `jikit` (*http://java.sun.com/products/jikit*) để quản lý việc tạo lập và duy trì các bó tài nguyên

Bước 4: Bản địa hoá các giao diện đồ hoạ

Bước cuối cùng chúng ta cần thực hiện là phải thể hiện ngôn ngữ bản địa ở các giao diện người dùng, Ví dụ "Cancel" : ở tiếng Anh, "Huỷ bỏ" ở tiếng Việt; vậy ta phải thể hiện được những giao diện bản địa hoá.

3.5 Khai thác các hàm của gettext

Chúng tôi xin được giới thiệu vắn tắt một số công cụ, hàm thường dùng của `gettext`:

Tên hàm	Chức năng
Textdomain	*định nghĩa tên dự án thông điệp*
Bindtextdomain	*Xác định đường dẫn, giá trị là chỉ định thư mục làm việc*
Gettext	*Tìm thông điệp nguồn msgid*
Dgettext	*Tìm thông điệp nguồn msgid ở textdomain nếu không thiết lập tên domain thì dgettext=gettext*
Dcgettext	*giống dgettext nhưng thêm đối số LC_MESSAGES*
Ngettext	*Các chức năng Ngettext,Dngettext,Dcngettext tương đương với gettext,Dgettext,dcgettext nhưng nó áp dung cho hình thức số nhiều (n=1 thì trả về msgid1, còn khác thì trả về msgid2)*
Dngettext	
Dcngettext	
bind_textdomain_codeset()	*Trả về chuỗi chứa codeset được chọn*

Các hàm này cần thiết cho quá trình bản địa hoá các thông điệp. Hai macro là ("xâu ký tự") và N_("xâu ký tự") thực chất là lời gọ i đến hàm gettext().

4. Thử nghiệm

Để thử nghiệm gettext, chúng tôi đã xây dựng một phần mềm đa ngữ để quản lý hóa đơn tiền điện. Với phần mềm này, người sử dụng có thể chọn lựa ngôn ngữ sử dụng là tiếng Việt hoặc tiếng Anh.

Trong phần mềm này, tất cả các biểu mẫu nhập số liệu, hóa đơn, báo cáo… đều thể hiện đúng trong ngôn ngữ được chọn. Ví dụ, mẫu đăng ký sử dụng điện:

MẪU ĐĂNG KÝ

Họ và tên:
Địa chỉ:
Số người dùng chung một công tơ:
Loại công tơ (1 giá, 3 giá):
Số kỳ ghi điện (1,2,3) :

REGISTRATION FORM

Name:
Address:
Number of peoples use the same meter:
Type of meter (1price, 3 prices):
Number of reading times (1,2,3):

Với phần mềm này, mỗi khi thay đổi về công thức tính toán, giá điện… chúng tôi chỉ cần sửa đổi mã nguồn chung của phần mềm mà không cần sửa ở các tập dữ liệu tài nguyên ngôn ngữ.

5. Kết luận

Việc sử dụng Gettext để đa ngữ hóa các phần mềm giúp cho việc đa ngữ được thực hiện nhanh chóng và dễ dàng hơn. Ngoài ra, nó cho phép chúng ta nâng cấp phần mềm một cách dễ dàng và ít sửa đổi nhất trên các mã nguồn chương trình.

Ngoài việc ứng dụng trực tiếp Gettext để đa ngữ hóa phần mềm, chúng ta có thể học hỏi được nhiều kinh nghiệm về cách thức tổ chức phần mềm, quản lý dữ liệu đa ngữ và phương pháp hỗ trợ dịch các thông điệp trong phần mềm dễ dàng nhất.

Tuy nhiên, công cụ gettext chỉ mới dừng ở mức độ hộc trợ đa ngữ hóa các thông điệp mà chưa hỗ trợ quản lý các biến đổi bên trong như yếu tố về văn hóa, các biến đổi đặc trưng cho từng ngôn ngữ. Trong tương lai, chúng tôi sẽ tiếp tục nghiên cứu để bổ sung vào gettext (vì đậy là phần mềm mã nguồn mở) một số hàm nhằm phục vụ xử lý ngôn ngữ như dò lỗi chính tả, sắp xếp, tìm kiếm, các tùy biến người dùng về màu sắc, hình ảnh…

TÀI LIỆU THAM KHẢO

[1] Võ Trung Hùng và Christian Boitet : “GetAMsg, une librairie pour le traitement de messages avec variantes et leur localisation”, Hội nghị quốc tế CIDE-8, Beyrut, Li Băng, 5/2005 (tiếng Pháp).

[2] Christian Boitet, Message Automata for Messages with Variants, and Methods for their Translation, Proceeding of the CICLING 2005, Mexico, 2-2005, Springer LNCS 3406, các trang 352—371.

[3] U. Drepper, J. Meyering, F. Pinard, B. Haible, GNU gettext tools, version 0.11.2, Published by the Free Software Foundation, 4-2002.

[4] IBM Corporation, International Components for Unicode (ICU) – User’s Guide, html:// oss.software.ibm.com/icu/userguide, 2003.

[5] Sun Microsystems Inc., Building International Applications, Sun product documentation http://docs.sun.com/db/doc/806-6663-01.

[6] Jean-Marc Hardy : “Multilinguisme : arme efficace mais lourde à manier”, Editions Dunod, Collection Planète numérique, page 88, 2004