

医学图像处理平台 3D Slicer 的国际化方法^①

曾文晔, 方路平, 万铮结

(浙江工业大学 信息工程学院, 杭州 310023)

摘要: 医学图像处理平台 3D Slicer 由于其免费开源, 扩展性强, 移植性好, 成为研究医学图像的良好平台, 受到世界范围内的研究者的关注. 根据 3D Slicer 的界面结构, 将其分为核心部分, 命令行模块, Python 脚本模块三部分进行分析, 利用 QT 国际化的工具, 分别进行国际化方法的研究, 找到了合适的国际化方法. 此外还对软件平台的字符串输入的过程进行分析, 通过对字符串编码的研究, 完成了该软件输入的国际化.

关键词: 3D Slicer; GUI; 国际化; QT; 编码

Internationalization of Medical Image Processing Platform 3D Slicer

ZENG Wen-Ye, FANG Lu-Ping, WAN Zheng-Jie

(Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: 3D Slicer, a useful Medical Image Processing Platform to research medical image, has obtained the attention of researchers worldwide. This platform has many advantages such as free and open source, high scalability, good portability etc. In this paper, 3D Slicer's GUI is analyzed and divided into three parts: the core, the command-line modules and the Python scripted modules. In order to achieve the goal of internationalization, QT's internationalization method is also researched in this paper according to different parts. Beside, the internationalization of input of 3D Slicer was also accomplished by analyzing the encoding of the input strings.

Key words: 3D Slicer; GUI; internationalization; QT; encoding

随着计算机技术的发展, 利用计算机对医学影像设备采集到的图像进行处理与分析的技术发展迅速, 广泛用于临床, 辅助医生进行清晰、准确的诊断及鉴别诊断^[1]. 一些医学图像软件平台也应运而生, 但是这些软件平台大部分是非开源的, 并且功能不齐或者扩展性能较差, 不利于我们的学习研究和二次开发, 而 3D Slicer 软件平台却是一个免费开源的医学图像处理平台. 对该平台的研究, 不仅对推广医学图像处理技术大有益处, 还对开发完善我们国家自己的医学图像处理平台有着重要意义.

1 3D Slicer 介绍^[2]

3D Slicer 项目是由美国波士顿布里格姆妇女医院手术计划实验室和麻省理工学院的人工智能实验室在 1998 年联合发起的, 其目的是开发一个易于使用的可

视化和分析的软件, 在 2007 年成功推出了具有完整体系结构的 Slicer 版本, 而最新版本是 2012 年发布的 3D Slicer 4.2.

3D Slicer 是一个免费, 开源的可视化和图像分析的软件平台, 它基于 VTK、ITK、Teem、QT 等开源软件的基础上开发而来的, 并支持多操作系统. 该平台的功能完善, 提供了图像分割, 配准和多模态图像数据的三维可视化功能, 对弥散张量成像的先进的图像分析算法, 磁共振成像和图像引导放射治疗功能, 支持标准图像文件格式, 集成与其他生物医学研究软件通信的接口. 除此之外, 3D Slicer 平台还对功能进行模块化设计, 其模块采用“plugin-in”方式, 集成到平台中, 方便功能的扩展和优化.

但 3D Slicer 平台在开发过程中, 只对英文这个单一语种提供了良好的支持, 尚未考虑到其他国家的使

① 收稿时间:2012-12-11;收到修改稿时间:2013-01-11

用者和开发人员,不符合软件国际化的要求.随着国际交流的密切,已成为其国际化推广和使用的瓶颈.

2 3D Slicer国际化方法介绍 和GUI分析

目前,软件国际化一般采用两种方式,第一种是直接源代码中直接使用目标语言,在切换不同语言版本的时候需要重新编译,这种方式由于需要重新编译程序,针对每个语言版本,都需要进行开发,效率低下.而第二种方法是对程序中用户可见的字符串使用翻译函数,动态地对这些字符串进行加载.程序开发人员只需要做好每种语言的资源包,即可实现多语言的程序版本.显然,第二种方法更为符合 3D Slicer 国际化的要求^[3].

3D Slicer 在版本 4 之前,其图像用户界面(GUI, Graphical User Interface)使用 Kitware 公司的 GUI 工具 Kwwidgets 开发,版本 4 使用了 Nokia 公司的 QT 代替了 Kwwidgets^[4].QT 对软件的国际化有着良好的支持,不仅提供了函数 `QObject::tr()` 和 `QCoreApplication::translate()` 来对目标字符串进行翻译,还提供了 `Lupdate`, `Linguist`, 和 `Lrelease` 三个工具来帮助开发者进行资源文件的生成.使用 QT 进行软件开发的用户如果想要让应用程序能使用多种语言,只需要做以下几步^[5]:

A. 确保每一个用户可见的字符串都使用了 `tr()` 函数或 `translate()` 封装,这两个函数的作用是将其封装的字符串文本进行翻译.翻译的过程中有两个比较重要的参数:上下文和源文.源文指的是需要进行翻译的字符串,上下文指的是源文所处的语言环境,以区分相同字符串在不同语言环境下的表达的意思的不同.一般的, `tr()` 函数是以调用它的 `QObject` 类或子类的名称作为上下文,而 `translate()` 函数则可以指定字符串作为上下文.

B. 通过工具(比如 `Lupdate`),将这些封装好的字符串,根据上下文的不同,写入一个以“ts”为后缀名的文件,这个文件是 XML 格式的.然后使用 `Linguist` 工具将字符串手动翻译,并使用 `Lrelease` 工具从 ts 文件生成一个 qm 文件,这个 qm 文件包含了程序所有可见字符串的翻译信息.

C. 最后在程序中使用 `QTranslator` 类安装 qm 翻译文件,翻译文件必须在相应字符串调用之前进行加载,一般的,翻译文件的加载在主函数中界面创建之前加载.

3D Slicer 目前的 GUI,根据其逻辑结构,可以分为三个主要部分:核心(the core),命令行模块(command-line modules),Python 脚本模块,而开发人员据此进行相应的 GUI 开发^[6].

对 3DSlicer 进行国际化,根据 GUI 的结构进行,需要进行的主要工作是字符串的封装和源文的提取和翻译两部分.最后加载翻译完成的 qm 文件.

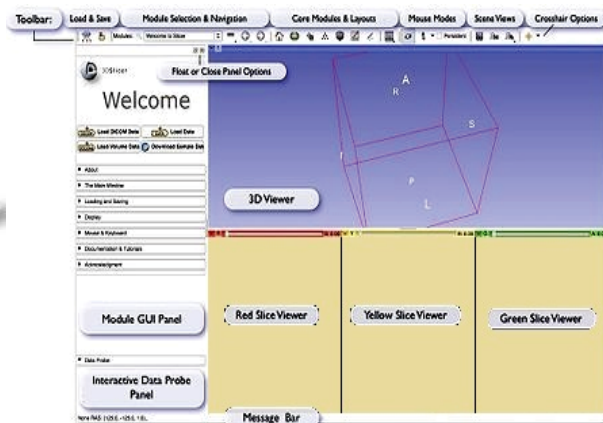


图 1 3D Slicer 的界面结构

3 核心部分的国际化过程

核心部分包括程序主窗口(Main Window), 数据探测模块(Data Probe), 以及核心模块(Core Module), 3D 视图窗口(3D Viewer)等.

程序首先创建一个主窗口,其中包含程序应用菜单,和部分工具栏,用于实现程序的一些基本操作,例如打开关闭文件,保存文件,程序基本设置等.然后在主窗口的基础上,添加部分自定义的窗口部件(Widget),这些窗口部件主要包括一部分自定义的工具栏,以及其他的窗口部件,比如 3D 视图窗口(3D Viewer).最后创建的是核心模块(Core Module),包括 Colors, Data, Models 等.

3.1 主窗口和核心模块

3D Slicer 核心的国际化首先是主窗口,主窗口部分的 GUI 的创建通过 QT Designer 工具图形化创建一个 UI 文件来完成,GUI 中出现的字符串都很好使用了 `QCoreApplication::translate()` 封装,只需要使用 `Lupdate` 工具将其中需要翻译的字符串提取即可.

核心模块的创建和主窗口类似,区别是核心模块的 GUI 还使用了一些由 CTK(Common Toolkit)中定义的窗口部件(Widget),在国际化的时候需要将 CTK 中

窗口部件(Widgets)包含的字符串也使用 Lupdate 提取。

3.2 工具栏

自定义的工具栏一共有 Module Selector(模块选择), Modules(模块快捷方式), Mouse Mode(鼠标模式), Capture(捕获), Viewers(视图中心)等。其中 Capture 工具栏在工程 qMRMLWidget 中定义, 而其余的工具栏都在工程 qSlicerBaseQTGUI 中创建, 以 Mouse Mode 工具栏为例说明工具栏的国际化过程。

在工程 qSlicerBaseQTGUI 中的文件 qSlicerMouseModeToolBar.cxx, 该文件的作用是创建 MouseMode 工具栏并实现其功能。其中使用了 setText() 和 setTooltip() 函数, 来实现按钮名称和提示信息。

```
this->CreateAndPlaceToolButton->setTooltip("Create and Place");
```

```
this->CreateAndPlaceToolButton->setText("Place");
```

这两行代码实现对象名为“CreateAndPlaceToolButton”的按钮的名称和提示信息的赋值。对这些用户可见字符串使用翻译函数, 然后使用 Lupdate 工具提取源文。最终完成后的效果如下图 2 所示, 以中文显示为例。

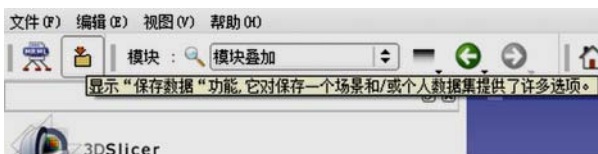


图 2 主窗口国际化效果图(中文)

4 命令行模块的国际化过程

所有命令行(CLI)模块都是由一个 QT 定义的框架, 然后在这个框架中引入许多个 Widgets 组合而成的, 所以命令行模块的界面的格式都是大致统一的, 图 3 以模块 Cast Image 为例说明。

如图所示, CLI 模块的公共部分有三块: qSlicerModulePanel, CLIProgressBar 以及 CLIModuleDialog, 如图 3 所示, 其中 CLIProgressBar 部件用于显示图像处理状态, 通过调用 VTK 库中的方法来实现, 在工程 qSlicerBaseQTCLI 中的 qSlicerCLIModuleWidget.cxx 文件中定义, qSlicerModulePanel 用于显示一般模块信息 CLIModuleDialog 提供应用/取消按钮进行图像处理控制, 分别由各自的 UI 文件 qSlicerModulePanel.ui 和 qSlicerCLIModuleWidget.ui 定义。它们的国际化方法和主窗口类似。只需对用户可见的字符串或字符串变

量使用翻译函数, 然后使用 QT 国际化工具, 得到 ts 文件, 翻译后便可获得相应的资源文件, 加载后即可。

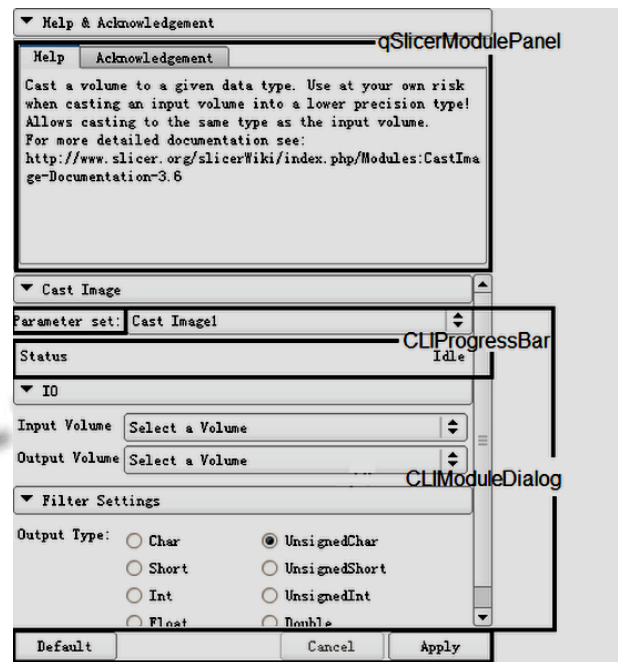


图 3 CLI 模块 Cast Image 界面结构

其余便是各个模块的私有部分, 其中使用到的 Widgets 的类在工程 qMRMLWidget 和 CTK 库定义, 这些 Widget 的属性值, 比如下拉菜单名, 标签名等, 在创建不同 CLI Module 时根据不同的模块做出更改。

4.1 模块界面

3D Slicer4 在创建完 CLI 模块的公共初始界面后, 通过添加 Widgets 的方法去完善这个界面。模块独立界面中 Widget 的类型和属性都是由各个模块的 XML 文档定义的, XML 文件中相应标签中的数据是创建模块大部分界面和功能的关键。

程序在编译创建 CLI 模块的过程中, 利用外部组件 SlicerExecutionModel 中的工具 GenerateCLP.exe 和库文件 ModuleDescription Parser.dll。前者用来在编译过程中处理 XML 文件, 从 XML 文件产生固定内容的头文件, 这个头文件中的一个字符串变量包含了 XML 中的所有信息。后者提供了一个类, 将这个字符串变量中包含的字符串分析, 并且存储在这个类的对象的属性中等待调用。

在 qSlicerCLIModuleWidget.cxx 文件中的函数 qSlicerCLIModuleWidgetPrivate::setupUi() 完成了整个模块界面的创建过程, 程序首先利用 UI 文件帮助创建

一个原始的界面框架，然后再通过函数 `addParameterGroups()` 将模块的一些控件加入到界面中去，来帮助完善界面和功能。`addParameterGroups()` 函数通过循环，将所有参数以及参数下面的具体项目分层次的全部添加到界面。

在添加具体的 Widget 的时候，给这个 Widget 的属性赋值，在赋值的时候将这个值进行国际化，那么在界面上就能显示希望的字符串语言。以给折叠按钮赋值的代码为例：

```
This->ModuleCollapsibleButton->setText(this->Title);
```

这句代码的功能是给 UI 文件中定义的模块折叠按钮 `ModuleCollapsibleButton` 的添加名称，这个按钮的功能是隐藏/显示整个模块，“`this->Title`”便是通过 `ModuleDescriptionParser` 来获取相应字符串的过程，将被定义为这个按钮的名称，如果将其进行国际化，那么就可以将界面中模块折叠按钮修改为其他语言。将源代码改为：

```
QString title_QS = this->Title;
const char *title = title_QS.toLatin1().data();
this->ModuleCollapsibleButton->setText(QCoreApplication::translate("title_QS",title));
```

前两行代码的作用是字符串的类型转换，因为 `this->Title` 返回的值的类型是 `QString` 而 `translate` 函数的输入参数类型是 `const char*`。最后一行代码进行翻译，上下文为 `title->Title` 的返回值，即是模块名。这样便可以使程序在此处加载翻译文件中相对应的字符串。

将 GUI 生成的过程中所有类似于模块折叠按钮的名称的赋值的函数，全部使用以上方式处理，那么只要有对应的 `qm` 文件，整个 GUI 就都可以国际化。

4.2 源文的获取

CLI 模块的用户可见的字符串都是通过变量传递的方式被使用，对相应变量使用翻译函数，就可以对所有的通过该变量传递的字符串产生翻译的作用。剩下的工作便仅仅是得到一个包含所有需要翻译的字符串的文件，并且翻译得到 `qm` 文件给程序进行加载。

由于 CLI 模块界面中字符串的值都是来源于 XML 文档，故获取其源文无法使用 QT 工具 `Lupdate` 来获取源文，需要重新编写一个工具来对 XML 中相应的字符串进行检索，实现 `Lupdate` 类似的功能。这个工具可以对文件夹及其子文件夹进行检索，找出所有的 XML 文件，然后对 XML 中的内容进行分析，检索出特定标

签中的字符串作为源文，模块名作为上下文。

为了实现上述几点，我们利用 QT Creator 开发平台，通过 `Qdir` 类实现文件夹的操作，通过 `QtXml` 类实现了 XML 文件的检索和最终需要的翻译文件的生成。最终完成国际化，效果如图 4 所示，以中文为例。

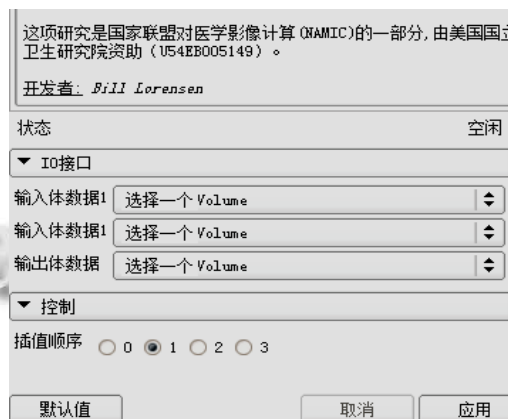


图 4 CLI 模块国际化效果图(中文)

5 Python 模块的国际化过程

3D Slicer 集成 Python 解析器，其大部分 API 都在 Python 中被封装。程序可以对定义脚本模块 (Scripted Modules) 的 PY 文件进行解析执行^[7]。复杂的交互模块可以完全通过 Python 来调用使用 Python 封装的 C++ 的方法来进行编写^[8] (例如：Endoscopy, Editor, SampleData, ChangeTracker 等)，即直接调用 QT 的类来创建界面，调用 VTK, ITK 等实现模块的功能。

因此 Python 脚本模块的国际化也和其他模块类似，只不过在调用翻译函数的方式有所区别，以 Endoscopy 模块为例，程序中用户可见的模块名字符串赋值的代码为：

```
parent.title = "Endoscopy"
使用 translate()函数，代码变为：
parent.title=
slicer.app.translate("Endoscopy","Endoscopy")
```

其中由于 `app` 是 Slicer 中类 `qSlicerApplication` 的对象，`qSlicerApplication` 是继承于 `QApplication` 的，故 `translate()` 函数可以采用上述方法调用。`Translate()` 函数的参数中，第一个参数是上下文，第二个是需要翻译的字符串。

由于 Python 文件的语法规则和编码方式都不同于一般 C++ 文件，无法通过 QT 的 `Lupdate` 工具进行源文的获

取. 故使用 Python 语言的 GUI 编程解决方案之一 PyQT4 中的工具 pylupdate4, 将 Python 文件中的被翻译函数封装的字符串的提取. 其中一个 Python 脚本模块 Endoscopy 国际化后部分界面效果如图 5, 以中文为例.

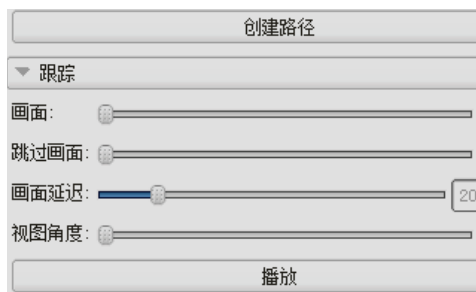


图 5 Endoscopy 脚本模块国际化后部分界面(中文)

6 字符串的输入

软件的国际化除界面以外, 还有字符串的输入, 比如, 打开和保存文件时的文件路径和名称等.

3D Slicer4 中的界面使用 QT 完成, 故供字符串输入的窗口部件都通过 QT 中的类来创建, 获取输入的字符串非常方便, 例如通过单行文本框 QLineEdit 类来获取用户输入的字符串时可以按如下方法进行:

```
QLineEdit *lineEdit = new QLineEdit;
QString Str = lineEdit->text();
```

在这两行代码中, 第一行代码创建单行文本框对象, 第二行代码将文本框中内容获取.

在 3D Slicer4 中, 用户输入的字符串首先通过上述方式在获取并在变量中存储, 当需要使用的时候, 在通过读取该变量的值来得到.

对字符串的存储和取出使用的过程进行分析, 发现存储在 QString 类型变量的字符串在转换为 const char* 类型变量使用时, 如果其转码过程使用了 toLatin1(), 则发生编码错误^[5], 中文字符全部转化为“?”, 而使用另一种方式 toAscii() 取代 toLatin1(), 并且在程序的 main 函数中将字符串的编码设置为“gbk”, 该问题得到解决. 3D Slicer 各处的字符串输入方式基本类似, 采用上述方式后用户可以使用任意语言输入文件名等信息. 相关 Web 页面获取模块: 以知识数据库中现有知识为基础, 根据 Web 站点配置信息生成动

态 URL 从互联网上获取与所需知识相关的 Web 页面.

7 结语

3DSlicer 既是一个医学图像的应用平台, 也是一个很好的学习研究平台. 该平台免费开源, 功能完善, 扩展性能优异, 给广大医学图像处理的研究者提供了种新的选择.

本文从 3DSlicer 的 GUI 结构入手, 将整个 GUI 分成核心, 命令行模块和 Python 脚本模块 3 个部分, 通过不同的场合灵活使用翻译函数, 实现了整个程序用户界面的国际化. 在资源文件的获取方面, 通过不同版本的 QT 国际化工具以及自行开发的 XML 检索程序, 实现资源文件的生成, 此外对平台的字符串输入部分, 分析了字符串的传递和编码过程, 并充分研究了 QT 中字符串类型转换的原理, 实现了字符串输入的国际化的国际化.

参考文献

- 1 田捷, 薛建, 戴亚康. 医学影像算法设计与平台构建. 北京: 清华大学出版社, 2007.
- 2 National Alliance for Medical Image Computing. Slicer4. [2011-10-09]. <http://www.slicer.org/slicerWiki/index.php/Slicer4>.
- 3 张阳, 蒋胜平, 高炜. 中文软件的国际化的方法研究. 电脑与信息技术, 2006, 12, 14(6).
- 4 National Alliance for Medical Image Computing. 3DSlicer [2012-08]. <http://en.wikipedia.org/wiki/3DSlicer>
- 5 Blanchette J, Summerfield M. C++ GUI QT4 编程. 北京: 电子工业出版社, 2010.
- 6 National Alliance for Medical Image Computing. Projects: ARRA: SlicerUI. [2011-11-10]. <http://www.na-mic.org/Wiki/index.php/Projects:ARRA:SlicerUI>
- 7 Pujol S, Pieper S. Programming in Slicer4. [2012-08] <http://wiki.Slicer.org/slicerWiki/index.php/Documentation/4.1/Training>
- 8 National Alliance for Medical Image Computing. Slicer4: Python. [2012-07]. <http://www.slicer.org/slicerWiki/index.php/Slicer4:Python>