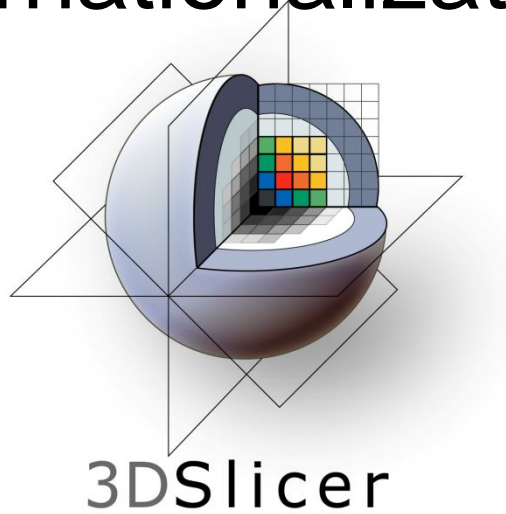


Internationalization



NA-MIC

25th Project Week June 2017



Overview

The language of a population is not localized by countries or states, because in most places more of one language is spoken and one language can be shared by more than one country or state. *Slicer 3D* is an *open source* and *cross platform* for medical image computing and visualization, an application widely used all over the world. A new feature for this application including the localization -*L10n*- and internationalization -*i18n*- capacities would facilitate its use to a wider number of speakers. This means its adaptation to different languages and regional differences.

This self contained project will focus on *i18n* also considering an approach to *medical terminology*. The process of software engineering pursues the specific aim of developing a prototype of the solution, including a basic understanding of the fundamental problem. There are different perspectives since the application has an extensive GUI: the internationalization of the core, the command-line modules and the python scripted modules. With the aim of adapting some modules of slicer 3D the current features of the software application and how they are implemented are to be analysed in order to design the first steps to a packaged solution.

Analysis

For the purpose of the Slicer 3D application to support multiple languages, the design of the structure and mechanism of the solution is to be significantly enhanced by the adoption of standards. Since the contributions to this open source software application come from an international community, the solution must be robust and easy to understand and apply. This software project does not need the user requirements or restrictions although to some degree it is necessary to recognize the needs of the translators and developers.

The focal point will be in design aspects: ***data encoding, data and documentation, software design, software development, and user interaction.***

For the ***data encoding***, our prime decision has been to study the use of the standard Unicode Transformation Format from Unicode implemented for XML as well as the Web Content Accessibility guidelines, regardless of the programming language used or the modules to be developed (CLI, python or C++).

Following the Web Content Accessibility guidelines, we must specify changes in the *natural* or human language used in documents. For this we must add *lang="Languagecode"* attribute to virtually any tag. It is also required to specify any change in language, this way we must declare the default, base or original language by adding *lang="Languagecode"* to the body or html tags. For this project we will be using the ISO639 code list. Corresponding to the ISO639-1 our aim would be:

- `body lang="en"` English
- `html lang="en-us"` English (United States variety)
- `html lang="en-gb"` English (Great Britain variety)
- `html lang="pt"` Portuguese
- `html lang="ar"` Arabic

- `html lang="fr"` French
- `html lang="es"` Spanish

For the ***data and documentation***, as well as how they will be used in the software development stage, there are several aspects to be studied, such as:

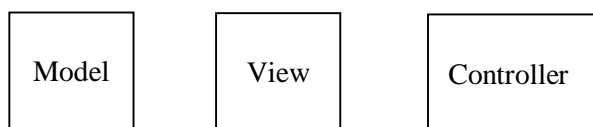
- The different sets of characters or symbols. We must ponder on Unicode standard to represent different languages with a single character encoding.
- The differences in character sizes and their impact in the layouts
- The writing direction, left-right in most European languages but right-left in Arabic.
- The different punctuation in different languages ((“ ”) in English - (« ») in French or (;) in Spanish – (؛) in Arabic)
- The keyboard shortcuts and their correspondence to a word in a particular language.
- The alteration of images and logos containing texts.
- Besides the translation of the existing text to other languages it is important to keep in mind that the files with parallel versions of texts are more difficult to maintain. For instance any change must be also made in all the translated versions.
- ...

The ***software design***, regarding the whole process of internationalization, will be based on the model view controller, a three-tier architecture pattern with its three tiers: presentation, logic and data tier.

In the first one, aka service layer, the front-end server will serve static and dynamic content because this will be the data to transfer to create the XML files. Is where the user will interact with the application. For this approach the XML files will be already provided to the module, so this interaction does not exist.

In the second one, aka business logic layer, the content from the front-end will be processed, here is where the rules are encoded for data to be created, stored and managed.

Finally the third one, aka persistence layer, where information will be stored and retrieved.



This project will be based in the rules implemented for the information to get from the languages files. How these rules are to be managed (the functions used or libraries) will be developed in future stages. This is to say: in the business logic layer. The software development will be implemented in python for an example module still to be determined. The files to store the different languages tags will be in XML format. According to this, we could determine future works.

In the ***software development*** stage, for this first test-project, it will be enough if we evaluate a small program according to the design. written in python and we make the changes in a small module to be later imported in 3D slicer. For future stages, we need to install the portable dependency libraries and to compile 3D slicer. If we think in the three-tier architecture, the main development will be in the

logic layer where the modules recover instructions from the interface module and consult the information in the files (model module) to send the translation of literals to the interface module.

Develop the prototype and check about the detailed architecture mode.

Check 3D slicer discussions

Qt libraries