

VoiceXML Translator

Mária Vámošová

Voice2Web Project

**R&D Centre for Mobile Applications (RDC)
Czech Technical University in Prague
Technicka 2, 166 27 Prague 6, Czech Republic**

Abstract

For my first project at RDC I was working on VoiceXML Translator. VoiceXML Translator is a utility for translating voice applications into another languages using web translation services. The main idea is, that user can create his VoiceXML in one language, then let it translate to another, and he can use both voice applications in both languages.

I focused on the main world languages - english, german, french and spanish, but the translator can translate to many others. The only problem is that some other languages, such as czech or slovak, aren't supported by voice server yet.

VoiceXML Translator is a part of Voice2Web project. This project is about the access and communication between the user and the internet via voice. For these voice applications the VoiceXML technology is used (will be described later).

For more information about Voice2Web project, visit <http://www.rdc.cz/en/projects/Voice2Web/>.

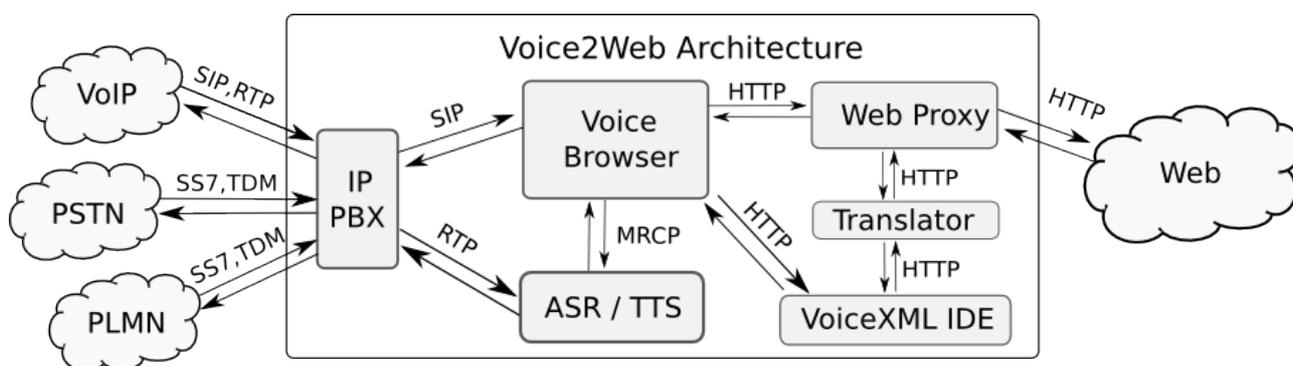
Introduction

At the beginning of my work, my goal was simple. I wanted a program, that can translate a voice application to other chosen languages, so the user can use both applications in both languages and writes only one. To translate a voice application, a VoiceXML needs to be translated.

The translation itself is done via web, using web translation services, because it's the most effective and simplest way when human translation isn't possible.

The translator is currently located at server *bolek* on *feld.cvut.cz*. It is accessible from VXMLIDE, a web service for creating and editing VoiceXMLs and therefore creating voice applications.

The architecture is following:



A caller from the VoIP, PSTN or PLMN network is authenticated and a connection is unified by the IP PBX. Unified internal VoIP connection (SIP and RTP) is processed by Voice Browser according to VoiceXML retrieved from VoiceXML IDE and with help of ASR and TTS engines. To enable effective WWW access, the Web Proxy intermediates the connection to the Web.

The Translator can be accessed from VoiceXML IDE and the web communication is done via Web Proxy.

Related Work

I'm surely not the only one who works on this problem. On the related problem worked my colleague Xavier García, whose work [\[1\]](#) I used in web translation services rating.

There are some works on the internet concerning the translation of whole VoiceXMLs.

One example is [\[2\]](#). They made a whole concept of translating the VoiceXML independent on the actual translation form (by machine or by human), based on XML representation of parts to translate.

My work isn't that complex. I focused on the translation via web, so I don't extract parts of VoiceXML that should be translated to an extra file, instead I change the text directly.

Other work I found, [\[3\]](#), is describing the problem of determining, which parts of VoiceXML should be translated. But compared to my work, they don't actually solve the translation itself.

Web translators rating

At the beginning it was very important to choose which web translation service is the best, so the translation would be of high quality. The criteria were:

- *number of combination of languages that translator can work with*
- *the quality of translation*

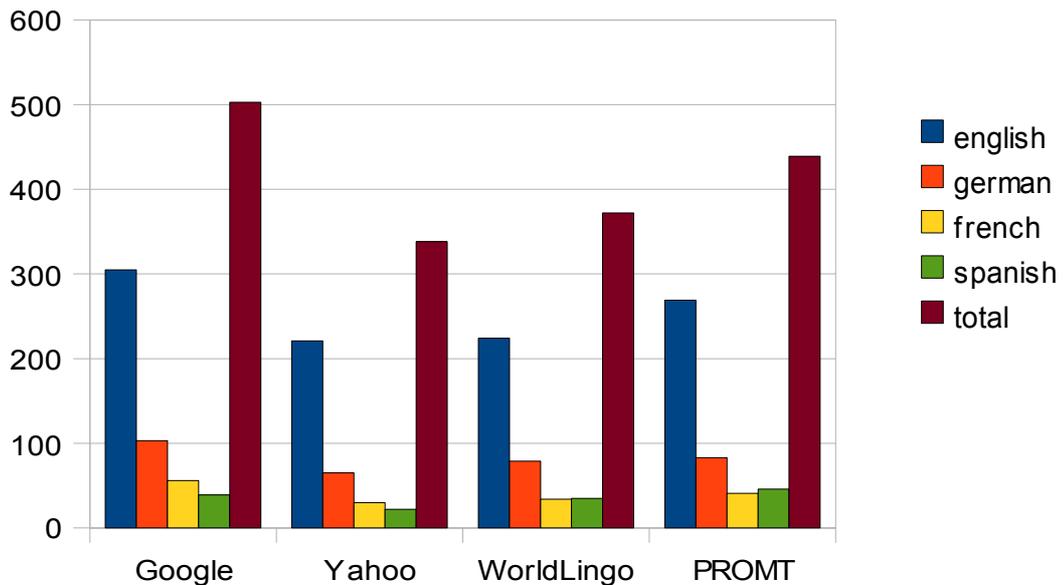
I researched available services, and found 4 services fulfilling my requirements:

1. Google Translate ([\[4\]](#))
2. Yahoo Babelfish ([\[5\]](#))
3. WorldLingo ([\[6\]](#))
4. PROMT ([\[7\]](#))

For the purpose of quality measurement, I picked 9 benchmark sentences in english from [\[1\]](#). I let them translate by experts or native speakers to 3 other languages - German, French and Spanish and also let the original english sentences to be corrected. Then I used web translators to translate these sentences, made a website and let people from RDC to rate all of these translations. Rater could give from 1 to 10 points to each set of translated sentences, and in the end, I summed the points of all translations from various languages to the one chosen language of each translator.

The results of this poll are following:

Provider	english	german	french	spanish	total
Google Translate	305	103	56	39	503
Yahoo Babelfish	221	65	30	22	338
WorldLingo	224	79	34	35	372
PROMT	269	83	41	46	439
Votes	19	7	3	3	32



Decription:

The table and graph show how many points reached the services in translations from various languages to the chosen language.

For my VoiceXML translation service I picked two translators with the highest score, namely Google Translate and PROMT.

VoiceXML Translation

VoiceXML is an XML standard for specifying interactive voice dialogues between a human and a computer. The specification can be found on the site [\[8\]](#). VoiceXML is interpreted by voice browser and is using tags for speech synthesis, speech recognition, and content all of these tags has to be translated.

After a deep study, I created a list of VoiceXML parts, that should be translated:

Text in tags, that are used for speech synthesis/speech recognition:

These tags are:

<audio>, <choice>, <item>, <option>, <paragraph>, <prompt>, <sentence>, <token>, <enumerate>, <prosody>, <rule>, <example>, <catch>, <block>

expr attribute

If it's a string value

message attribute

If it's a string value

alias attribute

If it's a string value

xml:lang attribute

That should be changed to new value

The Translator doesn't support the GSL formatted grammars, so the grammars need to be in XML format. This was done because of parsing problems, the GSL format isn't supported by standard XML-parsing libraries.

Software for VoiceXML Translation

User Interface:

VXML Translator consists of two public java classes - *Translator* and *TranslatorException*, that are stored in *cz.rdc.vxmlide* package. They can be used in other java programs, but can be also called directly. The *Translator* class takes several parameters:

```
Translator [input file] [output file] -tgt [destination language] -src [list of source languages separated by space] -v (optional)
```

Parameters description:

- tgt* parameter takes language specification in VXML format. The parameter specifies, in which language is the VXML going to be translated
- src* parameter takes a list of languages, that should be translated, separated by spaces. The languages should also be in VXML supported format.
- v* is an optional parameter and means that the program will also translate string variables in `<...expr=...>` and `<...message=...>` attributes.

Input file and output file can be omitted - the program will use standart input and output stream.

If there is any error during translation, the program will throw a *TranslatorException* and terminates.

Note that encoding should be in UTF-8 format.

Documentation:

The program is written as two java classes - *Translator* and *TranslatorException* in package *cz.rdc.vxmlide*.

The reason I picked java language for programming, is the compatibility with VXMLIDE, which is written also in java, and the presence of good XML and web communication libraries.

I picked C++ at first, but found out, that the libraries aren't good enough and the integration into IDE would be harder.

VoiceXML translator doesn't communicate directly with the web translators, instead it uses web proxy ([\[9\]](#)).

Via web proxy, the translator can communicate with different web translation services and use the same interface. The proxy also uses priority system - if the service with the highest priority isn't reachable, it tries to use the next one. The communication with the services is specified in web proxy configuration files.

The translator uses 2 web translation services, ordered by priority: Google and PROMT.

The communication with these services is up to web proxy. The configuration files are in web proxy home, in a folder named "VXMLTranslate".

Integration of VXML Translator to the VXMLIDE

Last part of my project assignment was the integration of VXML Translator to the VXML IDE - a web service for creating and editing VoiceXMLs.

This problem consist of two aspects:

- *changing the user interface*
- *creating a servlet*

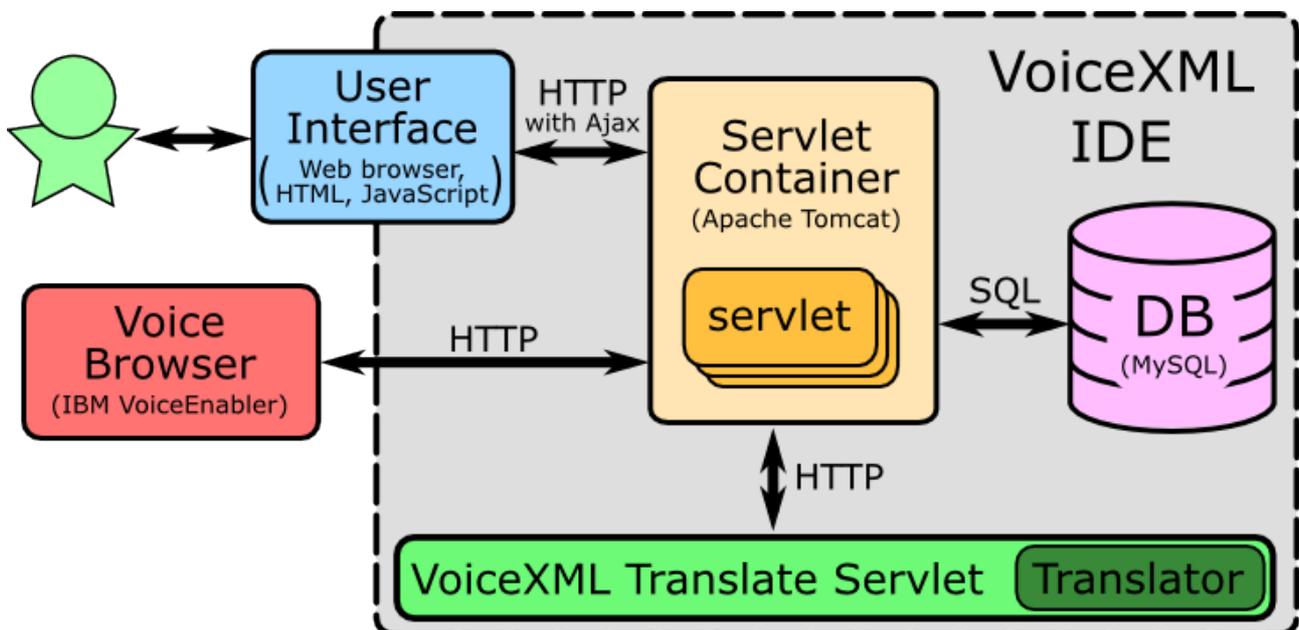
The interface part was very simple, I only made a form, which makes a request to the servlet part.

The servlet part consist of a java servlet class - *TranslatorServlet*, that takes 4 parameters:

- srcLang*: which language parts of VXML are going to be translated
- tgtLang*: in which language is the VXML going to be translated
- variables*: whether to translate string variables
- text*: the VXML itself as a string

The servet then creates an instance of *Translator* object and let the translator translate, setting the output stream parameter to the servlet's response stream.

The architecture of integration is following:



Conclusion

VoiceXML Translator can translate the text within the tags, that are used in speech synthesis and speech recognition and it can also translate the string values of *expr*, *message* and *alias* attributes of various tags.

It supports the choice of multiple source languages - it can translate only parts that are in chosen language and leaves the other parts untouched.

It's main advantage is, that it uses Google Translate as it's primary web translation service - this service is on it's rise, it's quality is highly rising and is probably the best machine translation service available.

The other important thing is, that it lets the user correct the translation and change something if he doesn't like it.

What the Translator can't deal with, are extern files, used in VoiceXML.

I thought it would be nice to let translate everything at once, but I found some problems:

- *the URI of translated file will be different and is not known at the moment of translation*
- *the program must have worked directly with the database*

Because of these problems the program doesn't translate extern files, but it can be done in the future.

The other problem is that it can't deal with grammar in GSL format. I think here is a great space for improvements.

Acknowledgement

We wish to thank IBM Research for generous support of the project and Jan Kleindienst, Borivoj Tydlit and Jan Sedivy for many thoughtfull suggestions.

References

- [1] Xavier García - Real Time Speech Translator Report
- [2] <http://www.speechtechmag.com/Articles/Column/Forward-Thinking/Extending-VoiceXML%27s-Impact-to-New-Markets-39760.aspx>
- [3] <http://docs.sun.com/app/docs/doc/820-1050/6ncof9ill?a=view>
- [4] <http://translate.google.com/>
- [5] <http://babelfish.yahoo.com/>
- [6] http://www.worldlingo.com/en/products_services/worldlingo_translator.html
- [7] http://www.online-translator.com/text_Translation.aspx
- [8] www.vxml.org
- [9] <http://bolek.feld.cvut.cz:8000/wiki/WebProxy>