Software Architecture for Better Text-based Information Accessibility

Vasile TOPAC\(^{\text{a}}\), Vasile STOICU-TIVADAR\(^{\text{b}}\)

\(^{\text{a}}\) Phd Student - Department of Computers and Information Technology
University “Politehnica” Timișoara
topacv@yahoo.com

\(^{\text{b}}\) Department of Automation and Applied Informatics
University “Politehnica” Timișoara
2 V. Pârvan blvd., Timișoara, 300223
ROMANIA
vasile.stoicu-tivadar@aut.upt.ro

Abstract

The paper suggests a software architecture to improve the accessibility to information from texts. This software combines techniques like: Image Processing, Optical Character Recognition, Machine Translation, Text Analyze and Text to Speech. The application uses a scanner or a web cam as an image input device, recognizes the text by using OCR, enables text translation by using Google’s machine translation implementation, interpret the text as a future development and reads the text by using TTS technology. In this way the user can put the text information source into a scanner or under a web cam and can hear the text translated and interpreted, if required. A functional prototype is presented and conclusions are issued.

1 Introduction

The XXI-st century was called "the century of information". Considering this, information accessibility is a big concern nowadays. Internet has done a huge step forward in this direction. Nevertheless the accessibility to information from text still has several limitations.

This paper presents a software project aimed to improve the accessibility to information from text. The possible limitations in accessibility are:

- vision problems (VP) when a person has low vision or blindness. In this case, the access to information from text, especially printed text without auxiliary help is very limited.
- language problems (LP) when a person has a text that is written in a language that he doesn’t understand.
- specialized text (SP) when a person reads a text written in a known language, but he can’t understand it because the text is specialized, containing many words specific to a specific domain (e.g. Medical diagnostic - can be unclear for non professional individuals).

Currently, one can find on the market products and applications that are addressing VP and LP problems, but neither of them is addressing all these problems, so there is no solution for the whole problem, but only for parts of it.

The currently identified solutions on the market are:
- Poet reader from Baum [1], that reads printed text. It’s a good tool for user affected by VP which can easily read the text from a printed paper.
- a mobile application named knfbReader Mobile [2] doing the same thing and working on a mobile device.
- an application for text recognition and translation for Android phones [3], but without text-to-speech capabilities.

Our purpose was to design and implement a software application that will be a complete solution for the accessibility problems previously described (VP, LP, SP).

Initially we designed the application only for VP and LP, and currently we work on the support for SP. The current prototype is capable of reading printed text, allowing the user to choose the language in which he wants to listen his printed text.

2 The software architecture

The current prototype architecture (SA1) is designed with respect to issues from [4]. There were several important decisions that we made in the planning phase:
- the application will not use any dedicated hardware; it should run on a PC, using common devices for image input (scanner, web cam or photo camera) and common speakers for speech synthesis
- the application will allow the user to run parts of the whole process chain (the user can use text-to-speech, or translation, or other combinations)
- integrate existing technologies: Optical Character Recognition - MODI, Translation – Google Translate API and TTS – Microsoft SAPI 5
- the application should let the user customize the speech synthesis by adding third party’s voices
- the addressed problems are only VP and LP related.

The developed program is a Windows application. It is created in Visual Studio 2008, based on the .NET Framework 3.5 and is written in C# language. It also consumes a web service provided by Google.
We considered that the application should be able to read printed text, translated into different languages. The process is presented in Figure 1:

The modules of SA1 are:
- image input (capable of grabbing images from a scanner, web cam or photo camera)
- optical character recognition (OCR), extracting text from an image
- text translation
- text-to-speech (TTS).

These modules are interconnected so that the information is successively transformed from printed text to digital text, translated text and at the end to speech. SA1 is illustrated in Figure 2.

The class view describing the project is presented in Figure 3.

To make the communication between modules more efficient we used interfaces, presented in Figure 4.

These correspond to the three modules of the application. They are used to separate the abstraction of the project from its implementation. This ensures that adding new functionality implementation does not require changes in the code on the superior level of the application. This is a Bridge design pattern [5].

The communication between the modules was optimized, so that if one module fails, the next module doesn’t wait for the response from the first one. This increases the application’s stability.

A deeper look into the modules is presented in the followings:

**Image capture module:** In this module we use
- **Direct Show API:** The Microsoft® DirectShow® application programming interface (API) [6] is a media-streaming architecture for the Microsoft Windows® platform. Using DirectShow, applications can perform high-quality video and audio playback or capture. We are using this when the input device is a web cam.

  and

- **Windows Image Acquisition (WIA),** being used when the image input device is a scanner or a photo camera.

**Optical Character Recognition module - MODI**
In this module we integrated a MODI component (Microsoft Office Document Imaging) [7]. It is a virtual printer driver for converting printed content into Microsoft Document Images or TIFF files. We only used it to perform optical character recognition (OCR). The interface for this module is presented in Figure 5.
For enabling translation, we used the Google translate API for .net. Google Translate [8] is a service provided by Google Inc. to translate a section of text, or a webpage, into another language, with limits to the number of paragraphs or range of technical terms, translated. For some languages, users are asked for alternate translations, such as for technical terms, to be included for future updates to the translation process. The interface for this module is displayed in Figure 6.

For performing text-to-speech, the application is using the Microsoft Speech API. SAPI 5.0 [9] is a programming standard that provides tools and components to speech recognition and text-to-speech software applications. Speech recognition applications allow a computer to respond to spoken commands or to write text in response to spoken words. Text-to-speech applications allow a computer to synthesize a voice that speaks words that are in a string or a text file. The interface for this module is displayed in Figure 7.

<table>
<thead>
<tr>
<th>System/Technology</th>
<th>Tool or Technology</th>
<th>Image capture module</th>
<th>OCR module</th>
<th>Text To Speech Module</th>
<th>Translate Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows Xp</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.NET Framework 3.5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Visual Studio 2008 (Visual C#)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DirectShow Lib 2005</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODI</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech API 5.0</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TTS Engine</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Google Translate API</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. – Technologies

The user interface is a Windows Form that enables the user to do the following actions:
- to preview the images taken by the web cam
- to grab images – by using the “capture” button
- to start the optical character recognition task – by using the “OCR” button
- to enable/disable translation and to chose the languages (from, to) for the translation.
- to choose the voice and language used by the speech synthesis
- to start the reading of the text – by using the “Read” button
- to make the user experience much easier, there is a button named “Do all” that initiates almost all the actions mentioned above.

The user interface design is presented in the Figure 8.
During the testing process, several lighting problems appeared. Due to the “chain architecture” of the application, the bad image has as a result a bad text recognition and the problem propagated to the end of the chain, making the synthesized speech result quite funny. As a conclusion, one would be careful with the used light and resolution.

After studying diverse use case scenarios, the next step was the improvement by adding new features:
- allowing text input not only from image or digital text, but also from speech, so that we will be able to get text from any media type possible (image, digital text, speech).
- text post editing (TPE) - interpreting and/or correcting translation of specialized text. By specialized text we understand text that has many words that are specific to certain fields, that can be difficult to understand for non-professional users, as an example a medical recommendation or diagnosis or legal/justice decisions.

The resulted architecture (SA2) is presented in Figure 9.

### 3 Text analysis methods

The first methods that we are considering are translation methods, because the interpretation task is very similar to translation, just that we are “translating” in the same language. **Phrase-based translation** (PBT) is a process of translating sequences of words to sequences of words, where the lengths can differ [10]. The sequences of words are called, for instance, blocks or phrases, but typically are not linguistic phrases but phrases found using statistical methods from a linguistic corpus. It was demonstrated that restricting the phrases to linguistic phrases decreases translation quality.

**Statistical Machine Translation** (SMT) is a machine translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. The document is translated with the probability $p(e | f)$ where a string $e$ in native language is the translation of a string $f$ in a foreign language. Generally, these probabilities are estimated using techniques of parameter estimation. The mathematics behind SMT mostly relies on the Bayes Theorem, which is applied to $p(e | f)$, the probability that the foreign string produces the native string to get

$$p(e | f) \propto p(f | e)p(e)$$

where the translation model $p(f | e)$ is the probability that the native string is the translation of the foreign string, and the language model $p(e)$ is the probability of seeing that native string. Mathematically, finding the best translation $\hat{e}$ is done by picking up the one that gives the highest probability:

$$\hat{e} = \arg\max_{e \in E} p(f | e)p(e)$$

For a rigorous implementation of this one would have to perform an exhaustive search by going through all strings $e$ in the native language. One of the problems that appear is that for SMT, the translation model is only able to translate small sequences of words and word order has to be taken into account somehow. Typical solution has been re-ordering models, where a distribution of location changes for each item of translation is approximated from aligned bi-text. Due to the differences between the particularities of fields of usage, SMT methods tends to split into fine-tuned version for each field (like [11]) assuring more accuracy in the interpretation.

**Fuzzy-matcher (FM)** is not a translation method, but it is a text analysis method that helps identifying similar phrases even if they have several differences. Using this method leads to store the phrase only in one form in the translation database, and identify all the variations around that phrase.

The approach that we consider as being the most accurate is to use a combination of PBT + FM or SMT + FM. By using this kind of combinations we can achieve a very good accuracy for interpretation. We studied a combination of SMT + FM [12], which was used for translation and had good results, which make us full confident that it will have the same result for text interpretation.
4 Conclusion

In this paper we proposed an application for improving the accessibility to information (from printed text or digital text or speech) by using advanced translation and text analysis methods. The project is under development, but we already have a functional prototype for SA1. After experiments, even if the accuracy is not 100%, the application proves that might be very helpful.

We identified several specific areas of application. As examples, several medical application areas (the TPE module will be implemented in the first phase for medical terms) are presented in the followings:

- **Empowerment of the patient** that assumes that the medical staff should inform as good as possible the patient about his/her medical state, and after the patient has a good understanding on this, he/she will have an important influence on the decisions that will concern him/her.

- **Evidence based medicine** - when medical staff communicates their latest experiences, so that others can profit of it. Usually this implies communication between medical staff speaking different languages.

- **Telemedicine** implying medical staff from different countries

Even started as a diploma project of the first of the authors of this article [13], this project continues as a PhD research. Nevertheless, the application has a very wide usage range, starting with reading books/documents or its translation for users that have vision problems or for any user that would need it. The application is innovative and has a great potential area of applicability.

5 Acknowledgements

We are grateful to Assoc. Prof. dr. Valentina E. Bălaş and Ing. Lucian Szantho (both from “Aurel Vlaicu” University of Arad) for their support and collaboration.

6 References