DadTEX — A full Arabic interface

Mustapha Eddahibi, Azzeddine Lazrek and Khalid Sami
Department of Computer Science,
Faculty of Science, University Cadi Ayyad
P.O. Box 2390, Marrakesh, Morocco
Phone: +212 24 43 46 49 Fax: +212 24 43 74 09
lazrek (at) ucam dot ac dot ma
http://www.ucam.ac.ma/fssm/rydarab

Abstract

This paper presents a TEX-based way to localize LATEX documents for natural languages with a script based on the Arabic alphabet. So, native speakers of such natural languages who do not know English can use and understand LATEX.

ضادتخ - واجهة عربية خالصة

ملخص: يعرض هذا المقال تعريبا كاملا لنظام تخ لتنضيد النصوص الشهير. يتيح هذا التعريب إمكانية تحرير النص المدخل بلغة عربية خالصة. الامر الذي يتيح للذين لا يستطيعون استعمال اللغة الانجليزية إمكانية استعمال برنامج ليتخ كنظرائهم والحصول على نصوص ذوات جودة تنضيد رفيعة على غرار ما يقدمه البرنامج الاصلي.

1 Introduction

Although the typesetting system TEX was originally designed especially for composition of mathematical documents, it has become a very fine system for typesetting documents in many other fields. The Arabic alphabet–based scripts are some of the linguistic contexts where the use of TEX has been progressively adapted. Many projects, including ArabTEX[1] [4], MlTEX,[2] Omega[3] [3], ε-TEX,[4] Aleph, and others, as multilingual systems, have been interested in typesetting documents containing simple Arabic text. The system RyDArab[5] has been dedicated to the composition of mathematical expressions in various notations used in Arabic, depending on the cultural context.

At first, TEX was intended for document composition in English. It was based on the ASCII encoding. To allow direct data entry for particular letters of Latin languages, a set of packages (`inputenc`, Babel,[6] ...) have been proposed. The application encTEX[7] [5], compatible with the standard 8-bit TEX, allows the encoding of input/output in UTF-8.

Initially, Arabic document composition was performed through Arabic/Latin transliterations. That is, the input of Arabic text was done with Latin characters corresponding to Arabic alphabet glyphs. The glyph transliterations are accompanied by the application of a set of contextual operations to find suitable glyphs. This is necessary because, besides its right-to-left writing direction, Arabic writing is cursive and letters have several shapes according to their positions in the word: initial, median, final, isolated (e.g., ج ـجـ ـج). The Omega and ArabTEX systems were interested in direct Arabic text input; i.e., text is directly typed in an Arabic text editor. This operation is performed using transliteration tables to translate Arabic text from the text editor encoding to the encoding used by the typesetting system.

[1] ArabTEX is a multilingual computer typesetting system developed by Klaus Lagally: http://www.informatik.uni-stuttgart.de/ifi/bs/research/arab_e.html

[2] MlTEX was developed by Michael Ferguson

[3] Omega is a 16-bit enhanced version of TEX developed by John Plaice and Yannis Haralambous: http://omega.enstb.org/

[4] ε-TEX is an extended TEX including the features of TEX--XET, developed as part of the $\mathcal{NTS}$ project by Peter Breitenlohner under the aegis of DANTE e.V. during 1992: http://www.ctan.org/tex-archive/systems/e-tex/

[5] RyDArab is a system for typesetting mathematics in an Arabic notation: http://www.ucam.ac.ma/fssm/rydarab/system/zip/rydarab.zip

[6] Babel provides multilingual support for TEX. It's developed by Johannes L. Braams: http://www.ctan.org/tex-archive/macros/latex/required/babel/babel.pdf

[7] encTEX allows full UTF-8 processing in standard 8-bit TEX. It's developed by Petr Olšák: http://www.olsak.net/enctex.html

Until now, however, there is no system that allows composing Arabic documents completely in Arabic. So, the development of a mechanism completely based on TEX to do so was an open question. Therefore, we developed an interface, called DadTEX,[8] that allows creation of LATEX documents in Arabic. The goal was to allow Arabic users, who don't know English, to compose LATEX documents containing *only* Arabic letters, with the addition of control characters like \, $, ... and of course with Latin text for bilingual documents.

Mathematical documents in an Arabic notation, composed with RyDArab and CurExt,[9] can so be typeset directly. In order to save the semantic meaning of mathematical commands and environments, it was natural to translate the RyDArab and CurExt macros. Of course, those translations are not enough. In TEX, besides the commands for mathematical objects, there are many commands that specify the mathematical content of the text, such as the `theorem` environment.

With these translations, Arabic users now hope that one need not use commands like `\chapter`, in a standard article. Clearly, if commands are also localized, it will be easy to understand their meaning; further, customization of command names allows TEX to be tailored to local pedagogical approaches.

Also, using English based commands with Arabic text is a source of ambiguity. Some problems with TEX source editing with bidirectional lines are:

Text selection: selections can be made either in logical or visual mode. Selections in logical mode are accompanied by discontinuities which is a direct consequence of bidirectionality and that leave the user feeling uncomfortable, while visual mode selections lead to content ambiguity.

Character positions: in a bidirectional line, the end and beginning of a run of Arabic text are visually in the same place.

Semantic ambiguity: characters without explicit direction inherit the direction of the preceding text. This misleads the reader of the source about the meaning of the expression. E.g., the command `\catcode‘\ب=11` will be displayed as the incorrect `\catcode‘\11=ب`.

With DadTEX,[10] such problems can be avoided by using only Arabic text and minimizing bidirectional text in source documents.

The RyDArab system has already made some steps in the way of TEX localization, by adding commands for automatic date generation with Arabic month names and Arabic numbers. Further steps to be done in this field include exploiting the linguistic and regional properties from the system settings and getting notational preferences (such as units, currency, ...).

2 DadTEX system

2.1 Process

At the beginning of this work, we thought of creating an application to convert Arabic text in a source document into its transliterated version using the same correspondences as those used in the transliterations. The program would read the source text character by character and translate each element into its equivalent. When a control character is met, the program would operate differently, by seeking in a commands dictionary the equivalent of this command.

This mechanism is similar to the one used in FarsiTEX[11] [1]. FarsiTEX translates Persian text to transliterated text via an application program `ftx2tex`. Commands are written with Latin characters but are rendered from right to left. For example, `\begin{document}` is written as `{document}begin\`.

This method, based on the use of an application to translate a localized source file, presents several drawbacks. For instance, it is not direct; it generates a supplementary file to be processed instead of the original source file. So, the compilation time is increased. Likewise, additional memory and free space are required. Therefore, the method adopted in DadTEX is based on a direct use of TEX through translation of commands into Arabic. The source file is compiled directly. This makes it possible to avoid the use of a supplemental application that could eventually limit the use of DadTEX across platforms.

Such command name translation is not sufficient when using some systems, such as RyDArab.

[8] Dad is an Arabic letter. Its sound doesn't exist in many languages, showing the Arabic language's phonetic precision. In the Arabic literature, the Arabic language is often called the Dad language.

[9] CurExt is a variable-sized curved symbols typesetting system: `http://www.ucam.ac.ma/fssm/rydarab/system/zip/curext.zip`

[10] DadTEX is developed within the framework of global project "Al-khawarizmy", acting in the general field of localization of e-documents and tools: `http://www.ucam.ac.ma/fssm/rydarab/system/zip/dadtex.zip`

[11] FarsiTEX is a Persian/English bidirectional typesetting system: `http://www.farsitex.org/`

Indeed, RyDArab is based on transliteration of individual letter-based alphabetical symbols [2]. Therefore, it becomes necessary to redefine the correspondences using Arabic characters. This operation has to take into account the particular encoding used.

Various systems of numbers are used in Arabic alphabet-based writings: the Arabic numbers notation system called Arabic or occidental numbers, used in North African Arabic countries; Arabic-Indic numbers, used in Middle Eastern Arabic countries; and Persian numbers, used in Iran. In TEX, numbers are used to control document component sizes, an action's frequency, etc. In this work, only the syntactic aspect of numbers will be discussed, the semantic would be in the scope of a future DadTEX version.

The method used in DadTEX is not specific to Arabic language, or languages with an Arabic alphabet–based script. It can be generalized on one hand to any right-to-left writing system, and on the other hand to any non-ASCII encoded documents. This system would alleviate the learning burden imposed on non-English speaking people.

2.2 Structure

For purposes of simplification, the first version of DadTEX is intended to be used with ArabTEX or Omega. Documents using DadTEX should be encoded either in the ISO-8859-6 encoding system for ArabTEX or UTF-8 for Omega. Other encoding systems can be used, as long as they are compatible with ArabTEX or Omega.

DadTEX is composed of four files:

`dadalpha.tex`: here, ISO-8859-6 Arabic characters are declared as letters using the primitive `\catcode`. In TEX, command names are composed of a control character \ followed by characters with category code equal to 11.

`dadbody.tex`: a set of commands used in document bodies are declared.

`dadpreamb.sty`: a set of commands used in document preambles are declared. The encoding system in use is defined here.

`dadadapt.tex`: DadTEX users can add their own commands. It is like a dictionary containing vocabulary that can be augmented whenever new keywords are established. Putting them in a separate dictionary allows for flexible translation.

DadTEX is platform independent; it can be used in both Windows and Linux. It is extensible, i.e., the user can add new commands. It is also flexible; all command names can be modified. The core of

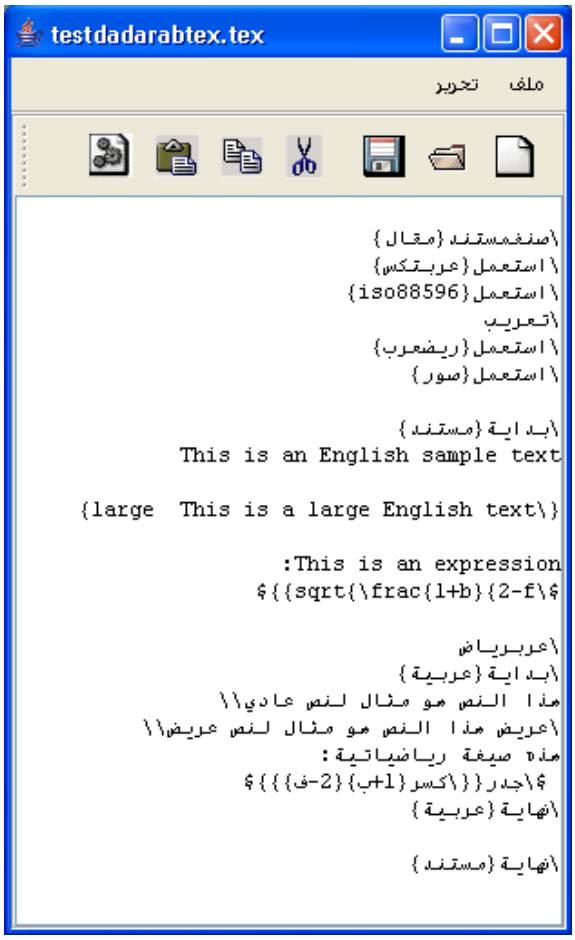

Figure 1: Arabic document source using ArabTEX

TEX and LATEX are not modified. In fact, DadTEX operates like an interface between the Arabic script user and the typesetting system. The mechanism used makes it possible to translate command names for all TEX packages available without limitations.

Figures 1–4 show examples of a TEX document both with and without DadTEX.

2.3 In action

Besides the DadTEX system, an adapted text editor, named DadNass,[12] has been developed to simplify typing of document source, calling TEX commands and running applications.

In spite of the Arabic interface for composition of TEX documents, some problems related to the debugger and to error messages remain. Error messages from the TEX core or from loaded packages are still displayed in English. Since DadTEX is based on

[12] `http://www.ucam.ac.ma/fssm/rydarab/system/zip/dadnass.zip`

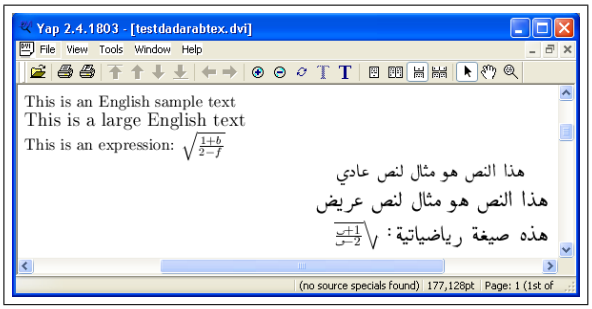

Figure 2: Output document in DVI

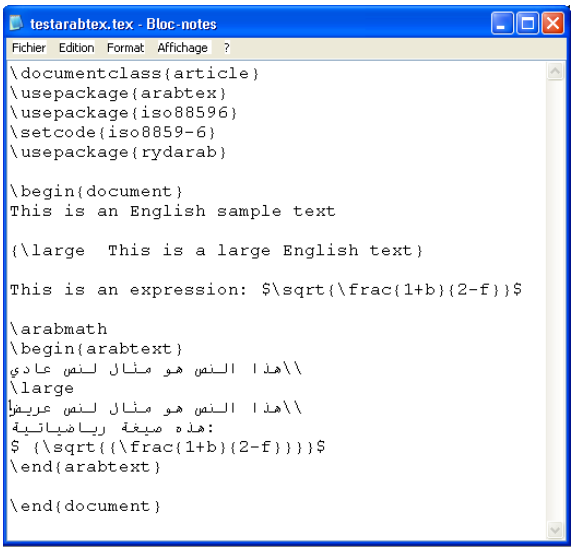

Figure 3: Normal equivalent of the document source

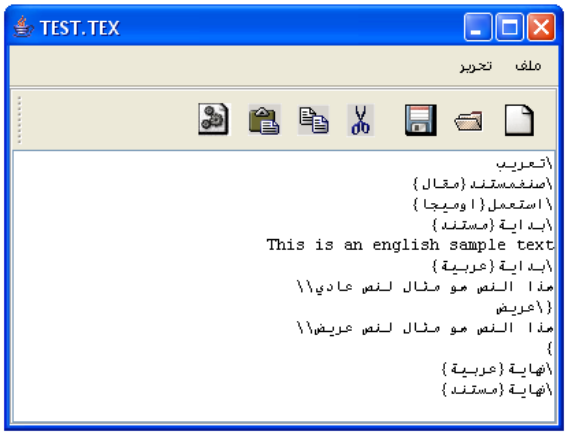

Figure 4: Simple Arabic document source using Omega

translation of command names only, error messages are always the same.

2.4 Encoding

DadTEX can be used with any encoding system capable of representing the Arabic alphabet, as long as the encoding is supported by the packages in use. In this version, we used ISO-8859-6 instead of UTF-8 because the ArabTEX system still has some problems when it is used with UTF-8. The encoding system recommended[13] to represent Arabic text under Unix is ISO-8859-6.

In ISO-8859-6, Arabic characters are encoded in one byte. It is thus easy to set their category code into 11. The assignment is done by using a text editor that supports ISO-8859-6 encoding (e.g., `\catcode‘\ب=11`). If the encoding used to encode Arabic text is UTF-8, then Arabic characters are presented using two bytes, and the use of a text editor that supports UTF-8 will lead to errors, because `\catcode` is intended to be used with only one-byte characters. For example, instead of using the command as follows `\catcode‘\ا=11`, characters should be divided into two visible bytes (Ø and § in the case of the Alef) using an ASCII text editor. Thus, we use the following: `\catcode‘\Ø=11` and `\catcode‘\§=11`. However, in the case of the Omega system, the hexadecimal code can be used directly: `\catcode‘^^^^0627=11`.

Currently available multilingual TEX typesetting systems require declaring a text's linguistic environment. For example, the Arabic text begins with the command `\begin{arab}` and ends with `\end{arab}`. This way allows TEX to use the suitable font and change the writing direction. This is in contrast with browsers, where no specific declaration of the language is required, thanks to an advanced level of use of Unicode's bidirectionality algorithm, and to the possibility of using available fonts instead of being tied to only one font. This is due to the multilingual nature of TrueType fonts. Indeed, all Unicode characters can be presented in a single TrueType font.

2.5 Document structure

The structure of DadTEX documents is similar to that used in LATEX.

The document starts with the Arabic command `\صنفمستند`, equivalent to `\documentclass` with the translation of the class name (e.g., `article` to

[13] “Arabization of graphical user interfaces”, by Franck Portaneri and Fethi Amara: `http://www.langbox.com/staff/arastub.html`

مقال). These classes and options specify the basic structure for a document.

Next, we can find commands like \استعمل, equivalent to the command `\usepackage`, which instructs LaTeX to load some external commands gathered in a package.

One can possibly use other commands that will be applied before `\begin{document}`, for example, commands that act on the document layout.

The command \بداية{مستند} is the Arabic translation of the command `\begin{document}` which marks the end of the preamble and declares, as its name indicates, that the document starts here.

The DadTeX system is not intended to be limited to Arabic script composition packages; all TeX commands can be translated. Indeed, a document can simultaneously contain text in Arabic and in other languages. For this reason, the beginning of the document is distinguished from the beginning of the Arabic environment. The command \بداية{عربية} is used to declare the beginning of the Arabic environment. For each command \بداية (`\begin`) there is a corresponding \نهاية (`\end`) command that indicates the end of the environment.

For the Omega-based DadTeX version, the first line of the document is not the Arabic equivalent of the command `\documentclass`. Instead, the Arabic source begins with the command \تعريب, equivalent to the commands that change ΩCP list, which will allow to conversion of the input document encoding into that used by the font, e.g., OT1 or T1. However, there are still some technical problems with using Omega and DadTeX with RyDArab to compose Arabic mathematical expressions.

2.6 Compilation

Instead of directly compiling the source document composed by the user, we use the combination of following compilation options:

```
latex \RequirePackage{adapt}
      \input myfile.tex -job-name myfile
```

3 Conclusion

DadTeX allows the TeX user who does not know English to use command names and parameters written in his mother tongue. So, TeX can be more user-friendly and understandable and, therefore, more widely used among Arabic-speaking people. Conversely, such users can thus be more concerned with and interested in TeX development.

In a future version of DadTeX, it will be very interesting to add support for the Arabi system [6]. Indeed, compared to ArabTeX, Arabi can be used with other packages with fewer restrictions.

Acknowledgements:
Thanks to Barbara Beeton and Karl Berry for their editorial corrections and contributions.

References

[1] Behdad Esfahbod and Roozbeh Pournader. FarsiTeX and the Iranian TeX Community. *TUGboat* 23:1, pages 41–45, Proceedings of the 2002 TUG Annual Meeting, Trivandrum, India, 2002.

[2] Mostafa Banouni, Azzeddine Lazrek et Khalid Sami. Une translittération arabe/roman pour un e-document. In *5e Colloque International sur le Document Électronique*, pages 123–137, Conférence Fédérative sur le Document, Hammamet, Tunisi, 2002.

[3] Yannis Haralambous and John Plaice. Multilingual Typesetting with Ω, a Case Study: Arabic. In *Proceedings of the International Symposium on Multilingual Information Processing*, pages 137–154, Tsukuba, 1997.

[4] Klaus Lagally. ArabTeX — Typesetting Arabic with vowels and ligatures. In *EuroTeX'92*, Brno, Czechoslovakia, 1992.

[5] Petr Olšák. Second Version of encTeX: UTF-8 Support. *TUGboat* 24:3, pages 499–501, Proceedings of the EuroTeX 2003 Meeting, Brest, France, 2003.

[6] Youssef Jabri. The Arabi system — TeX writes in Arabic and Farsi. In this volume, pp. 147–153.