Some Problems in Arabizing LaTeX

Klaus Lagally[1]
Universität Stuttgart, Institut für Informatik
Breitwiesenstraße 20-22, D-70565 Stuttgart, Germany
InterNet: lagally@informatik.uni-stuttgart.de

Abstract

ArabTeX, a macro package for use with TeX, the powerful typesetting system designed by D. E. Knuth, and LaTeX, its extension by L. Lamport to handle structured documents, was initially designed to support Arabic quotations inside scholarly texts written in some Western language. A prototype was released in the fall of 1991, and since then there have been two major revisions, extending the capabilities of the system far beyond the original goals.

The current version supports several different languages all using the Arabic script, and also a variety of different input notations. In this paper we discuss some of the problems which must be resolved in a multi-lingual system, and we present some local solutions as implemented inside ArabTeX.

Introduction

ArabTeX, a macro package for use with TeX[Knuth84], the powerful typesetting system designed by D. E. Knuth, and LaTeX[Lamport86], its extension by L. Lamport to handle structured documents, was initially designed to support Arabic quotations inside scholarly texts written in some Western language.

The first version of ArabTeX became operational in August 91, and as a consequence of operating experience and many suggestions from the user community Version 2 was released in May 92 [Lagally92a]. This version proved reasonably stable, and was presented, e.g. at the EuroTeX92 conference [Lagally92b] and at ICEMCO93 [Lagally92c]. Meanwhile many further user comments, suggestions, and requests for extensions reached us. We especially want to mention the remarkable contribution by Ivan Derzhanski, Edinburgh, who (without our prior knowledge) rewrote the module for the language-specific processing of Persian text, and who even contributed a new chapter to the user manual [Derzhanski93]. There were some other very reasonable suggestions for extensions, e.g. to provide for better formatting of poetry, correct handling of input text coded verbatim, and supporting the processing of existing machine readable input files encoded according to various standard conventions.

These new features could no more be easily fitted into the existing code, and thus a large portion of the system was substantially rewritten to produce Version 3 which was released in October 93, together with a new, expanded user manual [Lagally93].

In contrast to our original assumptions we realized during the development of Version 3, and especially while rewriting the documentation, that ArabTeX in fact is not really a system for supporting the writing of text in the *Arabic language,* but for supporting languages using the *Arabic script.* Insofar its name is misleading. In some respects it is a multi-lingual system, but this term has many different interpretations. This has to do with the fact that the term "language" may mean many different things to different people, especially if used in the wide sense that is customary

within Computer Science. A multi-lingual system has to do more than just providing some means of switching between different natural languages, and we have chosen to handle the various aspects separately. This turned out not only to be easier to implement, but also provided additional flexibility at no extra cost.

In this report we will discuss some of the various aspects that can be considered language dependent, and present the way they are presently handled within the ArabTEX system. Our concrete examples are displayed in a `typewriter` font.

ArabTEX Document Structure

An ArabTEX document is normally at least bilingual. It consists of a main text (usually) in some Western language (we shall call it the *primary language*) and insertions, or quotations, in some language using the Arabic script (called the *secondary language*). Insertions are bracketed by the command pair `\begin{arabtext}` and `\end{arabtext}` or, for very short quotations, by `<` and `>`, to indicate to TEX that special processing is required.

Both the primary and the secondary language may be changed independently within the document; we shall mainly discuss problems related to the secondary language, as the primary language issues can be handled by the standard TEX-LATEX mechanisms which do not concern us here.

If we use the term "language" in a wide sense, denoting any form of expression for certain features that can be changed independently, we end up with the following list:

- the language of markup commands,
- the encoding of input text,
- the natural language of the text,
- the output encoding (fonts),
- the global format of the document.

We shall now discuss these levels in some detail, and indicate how ArabTEX presently handles these matters.

The Markup Language

Any input file for TEX (or LATEX, or ArabTEX) contains two sorts of material:

- text to be typeset,
- markup commands, influencing the formatting by supplying semantic information on document structure and text interpretation.

The markup commands may be considered a language by themselves; a different set might be substituted without changing the structure of the document, and as TEX provides a macro definition facility this may even be done by using TEX's own mechanisms. In fact, this happens when using LATEX where many of the original TEX commands are replaced by different constructs. ArabTEX does the same, albeit to a lesser extent; there are also some additions for features not needed in European languages, e.g. spreading out a title across the whole width of the page.

The TEX mechanism for introducing new commands looks as follows:

`\def \`*command_name parameter_pattern* `{`*replacement text*`}`

LATEX provides an analogous mechanism denoted differently.

The Input Encoding

The input text has to be coded in some machine readable representation; the encoding for the primary and the secondary language need not be the same. The primary language is handled by TEX directly and thus has to be coded in ASCII or some of its 8-bit extensions in the ISO 8859-x family that is supported by TEX. There are some compatibility problems but these are outside the scope of this report.

The secondary language is handled by ArabTEX using its own reading routines; thus we obtain a high degree of flexibility. It is convenient to discuss the cases of 7-bit ASCII and other codes separately.

- 7-bit ASCII (ISO 646):

  This code is firmly standardized [ISO 646] and has been designed to be processed, transmitted, and displayed by virtually any computing system. As TEX expects its markup commands to be coded in ASCII, the latter is also the obvious first choice for the text to be typeset.

  Within ArabTEX this code is activated for the secondary language by `\setcode{arabtex}` or `\setcode{ascii}`, and it is also the default.

  Text in a foreign language can generally be represented in two different ways:

  – by a *transcription*, encoding the sounds of the spoken language, possibly with additional hints as to the correct spelling;
  – or by a *transliteration*, indicating the intended spelling directly.

  In Arabic, like e.g. in German, the two modes are rather closely related; English, or Chinese, are examples of languages where the transcription cannot be used without a fairly elaborate amount of linguistic information.

  Initially, ArabTEX supported only the Arabic *transcription* which is standardized ([DIN 31635], [ISO/R233]), and well known to orientalists. The spelling can be deduced from it without too much effort if some additional hints are given (e.g. for *tā' marbūṭa*, *'alif maqṣūra*, and *tanwīn*). We were not able to utilize the original notation for the standard transcription as it makes heavy use of diacritical marks outside of the standard ASCII character set; but we devised a very simple encoding that is still easily readable by humans (and could also be used for quite different purposes, as we noticed when we observed its use for electronic communication in a context where the text would never be processed by ArabTEX). Generating the standard notation for the transcription from this encoding is a nearly trivial task, as is shown in Figures 1 and 2.

  Input text coded in the transcription contains, and requires, the complete information on all vowels, and thus we can get the vowelized writing very easily; but we have to know the language sufficiently well to read it. Also, whenever we switch to a different language, e.g. Persian, we have to take into account that the transcription conventions may be different. For users not fluent in the language in question, and also for copying an unknown text, using a *transliteration* may be preferable. It may in fact be considered a special case, imagining a hypothetical language whose transcription rules are trivial, i.e. one-to-one. We may use the same ASCII encoding as above and get complete control of the spelling, including all diacritics.

```
\setcode{arabtex}
\setarab \vocalize

\centerline {<^gu.hA wa-.hamIruhu al-‘a^saraTu>}

\begin{arabtext}
i^star_A ^gu.hA ‘a^saraTa .hamIriN.
fari.ha bihA wa-sAqahA ’amAmahu,
_tumma rakiba wA.hidaN minhA.
wa-fI al-.t.tarIqi ‘adda .hamIrahu wa-huwa rAkibuN,
fa-wa^gadahA tis‘aTaN.
_tumma nazala wa-‘addahA fa-ra’AhA ‘a^saraTuN fa-qAla:

’am^sI wa-’aksibu .himAraN,
’af.dalu min ’an ’arkaba wa-’a_hsara .himAraN.
\end{arabtext}
```

Figure 1: Example of ASCII transcription encoding

ğuḥā wa-ḥamīruhu ’l-ᶜašaratu

ištarā ğuḥā ᶜašarata ḥamīrin. fariḥa bihā wa-sāqahā ᵓamāmahu, ṯumma rakiba wāḥidan minhā. wa-fī ’ṭ-ṭarīqi ᶜadda ḥamīrahu wa-huwa rākibun, fa-waǧadahā tisᶜatan. ṯumma nazala wa-ᶜaddahā fa-raᵓāhā ᶜašaratun fa-qāla:
ᵓamšī wa-ᵓaksibu ḥimāran, ᵓafḍalu min ᵓan ᵓarkaba wa-ᵓaḫsara ḥimāran.

Figure 2: Generated standard transcription

In ArabTEX a set of commands for choosing one of the language specific sets of transcription rules are provided: `\setarab`, `\setfarsi`, `\seturdu`, `\setpashto`, and `\setverb` to indicate one-to-one processing. These commands also have further effects on language specific processing, see below.

For an example of the resulting Arabic output, see Figure 3.

جُحَا وَحَمِيرُهُ العَشَرَةُ

اِشتَرَى جُحَا عَشَرَةَ حَمِيرٍ . فَرِحَ بِهَا وَسَاقَهَا أَمَامَهُ ، ثُمَّ رَكِبَ وَاحِدًا مِنهَا . وَفِي الطَّرِيقِ عَدَّ حَمِيرَهُ وَهُوَ رَاكِبٌ ، فَوَجَدَهَا تِسعَةً . ثُمَّ نَزَلَ وَعَدَّهَا فَرَآهَا عَشَرَةً فَقَالَ :

أَمشِي وَأَكسِبُ حِمَارًا ، أَفضَلُ مِن أَن أَركَبَ وَأَخسَرَ حِمَارًا .

Figure 3: Generated Arabic writing

- Other, non-ASCII, coding schemes:

  In some cases it may be desirable to process existing text files which have been produced using some other coding scheme, and which it may not be economically feasible to reencode. Generally the text in these files is coded verbatim, so we do not have to perform any language dependent processing; but we will usually not have sufficient information available to generate the transcription, and these files will also not be readable directly by a human without special "arabization software", and possibly also some special equipment.

  - ISO 8859-x:

    This family of codes contains 8-bit extensions for ASCII, assigning code positions for additional national characters used in various languages. They can be used with ArabTEX just like ASCII by using only the ASCII subset. Unfortunately, to our knowledge none of these codes contains all the characters required to code the standard transliteration directly, so we gain nothing by their use, but on the other hand we might run into difficulties if some equipment involved (e.g. some electronic mailing systems) cannot process 8-bit coded text correctly.

    ISO 8859-6 [ISO 8859-6] is an important special case: it contains the Arabic set of letters and some limited support for diacritics. Thus we can encode the transliteration directly if a bilingual text editor is available.

    In ArabTEX, this code is activated by `\setcode{iso8859-6}`.

  - ASMO 449 (ISO 9036):

    This is a 7-bit code ([ASMO 449], [ISO 9036]), replacing the ASCII letters with the Arabic letters, and otherwise nearly identical to ASCII (the exceptions do not concern us here). Like ISO 8859-6, it cannot be read directly without special software and equipment, but can safely be transmitted electronically.

    In ArabTEX, this code is activated by `\setcode{asmo449}`.

    For an example of ASMO 449 encoding, see Figure 4. Note that the commands are still coded in ASCII. The ensuing output is shown in Figure 5.

  - ISO 10646 and UNICODE:

    These codes are presently not supported by ArabTEX as TEX cannot directly handle them.

  - Further coding schemes:

    There are presently more than 10 coding schemes for Arabic in current use. As version 3 of ArabTEX provides a standard interface for additional reading modules, adding the power to process further encodings is just a clerical task. Work along these lines is in progress.

```
\setcode{asmo449}
\begin{arabtext}
gRjeIl SGMpbnIl eofpjnJr HpgG GdLnHgnIo GdESdGejqnIo
HfGRjQ JnNQoLo eofJnUpQnIk ef GdGfJpNGHGJp GdHGcSJGfjqnI
ESdGe GHGO - GdMjGI

engqnOnJ GdGfJpNGHGJo GdJqnTQjYpjqnIo ai HGcSJGf GdJi
LnQnJ GdCQHYGA dpYnhrOnIp HfGRjQ HhJh RnYjenIp MpRHp
GdTYHp GdHGcSJGfi Edi GdSqodWnIp , HnYOneG GfJnUnQnJ Ydi
eofGapSpgG fhGR TQja QFjSp GdhRQGAp GdSqGHpb.  haGRn
RnhLo HfGRjQ GUa RQOGQi HpSoghdnIm ai eobGWnYnIp GdSqpfO
GdLnfhHpjqnI ajeG SnbnWn TnbjbogG eQJVi GdPi dner Jncofr
QGVpjnIk Ynfgo.

EdG Cfqn HhJh dner JnfLnMr ai GdMoUhdp Ydi GdCZdnHpjqnIp
GdeoWdnbnIp ai GdHnQdneGfp GdLnOjOp.  hbnOqnenJ HnYrOn
EYdGfp fnJjLnIp GdGfJpNGHGJp TocQngG Edi GdCoeqnI GdJi
enfnMnJ MpRHn GdTqnYHp GdHGcSJGfi KpbnJngG.  EdG CfqngG
CVGanJ: "cofqG fnJnhnbqnYo Cfr jnchfn anhrRofG CcHnQn".
\end{arabtext}
```

Figure 4: Example of ASMO 449 encoding

The Natural Language of the Text

Processing conventions may depend on the natural language in which the text is written. The ensuing problems have been discussed extensively within the TEX user community, and in the context of European languages specific LATEX extensions are available, e.g. the Babel system [Braams93]. Among the points to be covered are:

- hyphenation of words,
- special punctuation rules,
- capitalizing conventions,
- handling of compound words,
- diacritics and special letters.

As these points concern the primary language, we will not discuss them here any further.

For the secondary languages related to Arabic some other language dependent conventions have to be observed; a list of examples includes:

- right-to-left writing (in all cases),
- special handling of numbers,
- ligature conventions,
- additional national characters, for e.g. Persian, Urdu, and Pashto,
- the collating sequence which is language dependent,

هزيمةٌ ساحِقَةٌ مُنِيَتْ بِها الجَبهَةُ الإسلاميَّةُ بنازير تَخرُجُ مُنتَصِرَةً من الانتِخاباتِ الباكستانيَّة إسلام اباد ـ الحياة

مَهَّدَت الانتِخاباتُ التَّشريعِيَّةُ فى باكستان التى جَرَت الأربعاء لِعَوْدَةِ بنازير بوتو زَعيمَةِ حِزبِ الشعبِ الباكستانى إلى السُّلطَةِ ، بَعدَما انتَصَرَت على مُنافِسِها نواز شريف رئيسِ الوزراءِ السّابِق . وفازَ زَوجُ بنازير اصف زردارى بِسُهولَةٍ فى مُقاطَعَةِ السِّند الجَنوبِيَّة فيما سَقَطَ شَقيقُها مرتضى الذى لَمْ تَكُنْ راضِيَةً عَنهُ .

إلا أنَّ بوتو لَمْ تَنجَحْ فى الحُصولِ على الأغلَبِيَّةِ المُطلَقَةِ فى البَرلَمانِ الجَديدِ . وقَدَّمَت بَعْدَ إعلانِ نَتيجَةِ الانتِخاباتِ شُكرَها إلى الأُمَّة التى مَنَحَت حِزبَ الشَّعبِ الباكستانى ثِقَتَها . إلا أنَّها أضافَت : «كُنّا نَتَوَقَّعُ أنْ يَكونَ فَوْزُنا أكبَرَ » .

Figure 5: Output of the ASMO 449 example

- elongating words for filling lines,
- transcription rules, according to standard conventions for the chosen language,
- *hamza* rules, silent *'alif*, for Arabic and related languages,
- *'izafet* rules, silent *hā'*, for Persian and related languages,
- positioning of *tanwīn fatḥa*, as there are language-dependent conventions,
- vowelization conventions,
- assimilation, of the article and also elsewhere.

Within ArabTEX, the secondary language is selected by one of the aforementioned commands `\setarab`, `\setfarsi`, `\seturdu`, `\setpashto`; the command `\setverb` will switch to the hypothetical "verbatim" language with no special processing at all.

Even inside a given natural language there may be variants, e.g. the historical vs. the modern conventions for the carrier of *hamza*. An extreme case in this respect is represented by the special notations and annotation conventions used in the Holy Qur'ān. In ArabTEX they are presently not properly handled, and in our opinion need not be, as we feel that this text rather ought to be written by a calligrapher. There is no general consent on this matter, and for the issues involved see, e.g. [Haralambous92], [Saba94], or the appendix of any Qur'ān edition.

The Output Encoding

This is usually considered just the issue of choosing an appropriate member of a set of fonts usable for the language in question. For the Arabic script there are some more options:

- Whereas some ligatures are mandatory, most of them are optional. ArabTEX will generate a large number of them automatically, but it is possible to turn them off individually or globally (with the exception of *lām-'alif*).
- If the text is input in the transcription mode the complete vowel information is available. Nonetheless we might want to, and can easily, control the generation of vowel marks locally and globally.

- If vowel marks are generated, they may be positioned, at the user's choice, all at the same level or at a position depending on the height of the base character. This is a matter of taste, and also depends on the text at hand.
- Titles may be required to be right adjusted, centered, or spread out across the available space by lengthening some connecting lines between letters. Spreading out may also be desirable in other circumstances, and can be controlled locally and globally.

Usually there are some language specific default conventions (e.g. the use of a Nasta'liq font for Persian), which we should be able to override locally or globally, and ArabTEX provides a rich complement of options which we will not describe here in detail.

The Global Document Format

The general layout of a document depends on the type of the document, the primary language, and possibly some local conventions, and is normally fixed globally by a "document style". In LaTeX the standard styles "article", "report", "book", and "letter" are provided, and meanwhile many customized variants exist according to local needs or special guidelines.

As ArabTEX presently presupposes the primary language to be European (including US-English), the normal LaTeX styles can be used. However, this may lead to awkward results if the document contains mostly (or exclusively) Arabic text. For documents following the Arabic conventions the obvious solution would be to write additional style files for Arabic as a primary language. This, however, is not a trivial task as many design decisions may be buried deeply inside the macros of LaTeX, or even inside the program of TEX itself where they cannot be changed without losing compatibility with standard applications.

These matters are currently under investigation; some of the relevant issues to consider are:

- format of page headers and footers,
- page numbering conventions,
- format of captions,
- table of contents, list of figures, list of tables,
- index, glossary, references list,
- lines ragged left vs. aligned,
- paragraph indentation,
- ordering of columns in the text,
- ordering of columns in tables,
- itemizing and enumerating in lists,
- format of footnotes,
- cross referencing and citations.

Conclusion

We have indicated that in a multi-lingual system some sort of language switching occurs at many different levels which are practically independent of each other. We are very skeptical about the existence of a single concept covering them all, and about the feasibility of a standard language switching mechanism.

Related work

TUG, the TEX users group, has installed several task forces concerned with the future development of LATEX, possible successors to TEX, and support packages. Some of their internal working documents, e.g. [Gaulle93], [Ziv93], and [Haralambous93], are concerned with multi-lingual issues. Whereas to our knowledge no definite proposal has yet emerged, and the general direction of the discussion is somewhat different from our approach, we gratefully acknowledge the strong influence these discussions had on the development of ArabTEX Version 3, and thus indirectly on this report.

Acknowledgments

The ideas presented in this report have grown out of discussions with many people. We are especially indebted to Alaaddin AlDhahir, Johannes Braams, Bernard Gaulle, Yannis Haralambous, and Alon Ziv of the electronic discussion list TWGMLC-L, to Jörg Mittelbach and Rainer Schöpf of the LATEX3 project, to Nicholas Heer, Dilworth Parkinson, Mohamed Saba, and Waheed Samy of the discussion list Arabic-L, and to the anonymous powers generously providing access to the InterNet.

References

[ASMO 449] ASMO 449: *7-bit coded Arabic character set for information interchange,* Arabic Standards and Measurements Organization, 1982.

[Braams93] Johannes Braams, *Babel, a multi-lingual style-option system for use with* LATEX*'s standard document styles,* Preprint, July 22, 1993.

[Derzhanski93] Ivan Derzhanski, *Some Contributions to the Parsing Routines and the ArabTEX User Manual,* (our heading), Personal communication, 1993.

[DIN 31635] DIN 31 635: *Umschrift des Arabischen Alphabets*, Deutsches Institut für Normung e.V., 1982.

[Gaulle93] Bernard Gaulle, *(ltx3) requirements for ML docs,* Memo to the LATEX3 5.4 research task group on "typographical conventions and requirements in multilingual environments", June 1993 (personal communication).

[Haralambous92] Yannis Haralambous, *Typesetting the Holy Qur'ān with TEX,* in: *Proceedings of the 3rd International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman Script),* University of Durham, UK, December 10-12, 1992.

[Haralambous93] Yannis HARALAMBOUS, *Proposal for a semi-standard TEX Arabic/Farsi/ Urdu/Pashto transliteration,* Memo to TWGMLC, TEX Users Group TEXnical Working Group on Multi-Language Cooperation, May 1993 (personal communication).

[ISO 646] ISO 646: *Information processing — ISO 7-bit coded character set for information interchange,* International Organization for Standardization.

[ISO 8859-6] ISO 8859-6: *Information processing — 8-bit single-byte coded graphic character sets — Part 6: Latin/Arabic alphabet,* International Organization for Standardization, 1987.

[ISO 9036] ISO 9036: *Information processing — Arabic 7-bit coded character set for information interchange,* International Organization for Standardization, 1987.

[ISO/R233] ISO/R 233 - 1961: *International System for the Transliteration of Arabic Characters*, International Standards Institution, 1961.

[Knuth84] Donald E. KNUTH, *The TEXbook*, Volume A of *Computers & Typesetting*, Addison-Wesley, Reading, Mass., 1984.

[Lagally92a] Klaus LAGALLY, *ArabTEX, a System for Typesetting Arabic, User Manual Version 2.05.* Report 1992/06, Fakultät Informatik, Universität Stuttgart, 1992 (outdated).

[Lagally92b] Klaus LAGALLY, *ArabTEX - Typesetting Arabic with Vowels and Ligatures*, in: *EuroTEX '92, Proceedings of the 7th European TEX Conference, Prague, Czechoslovakia, September 14-18, 1992.* Also available as: Report 1992/07, Fakultät Informatik, Universität Stuttgart, 1992.

[Lagally92c] Klaus LAGALLY, *ArabTEX, a System for Typesetting Arabic*, in: *Proceedings of the 3rd International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman Script), University of Durham, UK, December 10-12, 1992.* Also available as: Report 1992/11, Fakultät Informatik, Universität Stuttgart, 1992.

[Lagally93] Klaus LAGALLY, *ArabTEX, a System for Typesetting Arabic, User Manual Version 3.00.* Report 1993/11, Fakultät Informatik, Universität Stuttgart, 1993.

[Lamport86] Leslie LAMPORT, LATEX, *a Document Preparation System*, Addison-Wesley, Reading, Mass., 1986.

[Saba94] Mohamed SABA, *A proposal for an Arabic standard code for information interchange,* ICEMCO94, 4th International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman Script), London, 7-9 April 1994.

[Ziv93] Alon ZIV, *Bidirectional Printing in* LATEX*: Problems and Solutions*, Preprint, July 1993

[1] Klaus Lagally, born in Munich (Germany) in 1937, University studies in Mathematics and Physics, Ph.D. in Theoretical Physics 1967. Work on Operating Systems and Programming Languages, Professor of Computer Science 1976, Universität Stuttgart, Germany. Current research interests: Arabic text processing and typesetting.