A brief history of TeX, volume II

Arthur Reutenauer[*]
ENST Bretagne
Technopôle Brest-Iroise
CS 83818
29238 BREST CEDEX 3
arthur dot reutenauer (at) normalesup dot org

Rationale for this *volume II*

When I gave this talk in Bachotek I appended the subtitle *Pax TeXnica — the program on which the sun never sets* — an obvious pun to two historical empires renowned for their considerable geographical extent. I wanted to add this subtitle for two reasons: first, I liked *A brief history of TeX* a lot but I realized after choosing it that there had already been a talk with this exact same title more than ten years before[1] and I wanted to avoid the risk of confusion, be it only for archival purposes; and second, I felt the subtitle made my standpoint clear — the history I wanted to account for was very much a *geographical* one, namely how TeX enabled us to gradually typeset in every language of the world — or almost so. As far as the printed version was concerned, though, it seemed that it could also be considered a sequel to the first article — after all, many things have changed over ten years! Hence this *volume II*.

But first, let us recapitulate things from the beginning ...

1 The origins

1.1 In the beginning there was ...

History begins, scholars tell us, with the invention of writing and the ability to account for one's own culture. So, in the beginning there was typesetting and the program that enabled us to do so, let us call it TeX. This program was written by a man, let us call him 高德纳. Oh, and we need a date, too, so let's say 1978, thirty years ago.

高德纳, as the name suggests, lived in a region inhabited by many Chinese citizens, next to the great city that goes by the name of the Old Golden Hills. But he was an American citizen and a native speaker of the English language. So the program he wrote was all in English (with a lot of '\' though) and it was meant — at first — for English speakers to use.

Ööç

Figure 1: The name of a distant galaxy. From the very beginning, TeX sets out to conquer the universe (extract from `story.tex`, in *The TeXbook*, chapter 6).

Nevertheless, when 高德纳 created TeX, he still thought of the users speaking other languages. Of course, all the commands were in English, the default settings were chosen for that language, and the fonts used a 7-bit encoding[2] supporting only the Latin script, but he made provisions for extending this. The fonts, in particular, were supplied with a set of diacritics with the help of which he devised a set of well-known accent commands that could construct an accented character "on the fly", a sample of which can be seen in figure 1, an extract of a famous TeX file. This made it possible to write, mostly, all the languages of Western Europe — and therefore of all the other countries that use the same languages, in particular all of South America.

This was the first step since, even if they may seem impractical now, the accent commands actually introduced a way of inputting a lot of characters users didn't have access to on a standard American keyboard, much in the way math commands were a way of specifying the layout of complicated math formulæ; so even if they were not an encoding[3] in the current meaning of the term, they were a sort of coding system, and were thought as such by many users as well as some recoding utilities.[4]

So TeX extended, from the very beginning, over all the Americas as well as the western part of Europe,

[*] I wish to thank Jerzy Ludwichowski heartily for his constant encouragement to write this article and his patience in waiting for it.

[1] This talk was given in Toruń in 1995 and published the next year by different journals, including *TUGboat*, where it is available online: `http://www.tug.org/TUGboat/Articles/tb17-4/tb53tayl.pdf`.

[2] That is, they could only have up to $2^7 = 128$ characters.

[3] In the sense of an encoded character set, like ASCII, the ISO-8859-* family, or Unicode.

[4] To name just one, the very popular `recode` program (`ftp://ftp.gnu.org/pub/gnu/recode/`) knows about sequences likes `\'e` as the "TeX" encoding.

"Uznając, iż los nas wszystkich od ugruntowania i wydoskonalenia konstytucji narodowej jedynie zawisł, długim doświadczeniem poznawszy zadawnione rządu naszego wady, a chcąc korzystać z pory, w jakiej się Europa znajduje i z tej dogorywającej chwili ..."

Figure 2: Polish uses a lot of diacritics (extract from the May Third Constitution, 1791).

and many regions in Africa.[5]

But, as mentioned, this wasn't enough even for some other languages using the Latin alphabet, let alone languages using any other alphabet or different writing systems. Work needed to be done, as 高德纳 acknowledged that he couldn't handle all the languages of the world by himself, and he encouraged people to settle to this task. It wasn't long before people did so.

1.2 Go East

As TEX was born, the story tells us further, there was a companion program called METAFONT, whose purpose was to design the fonts that TEX used. As a matter of fact, all the letters and accents we discussed above were all drawn using METAFONT, so adapting the fonts to other languages meant, mostly, drawing more characters as needed.

Let's discuss how this was done. An interesting example is Polish. It uses a wealth of accents (see figure 2); most of them were already present in the fonts or easy to add, by simple modifications to the existing characters. One of these accents, though, is quite special: it's called *ogonek* which means "little tail" in Polish, as for example on the first word of figure 2. It looks remotely like a reversed cedilla but is not quite, and the drawing had therefore to be invented from scratch and polished carefully.[6] Then, a new control sequence had to be invented and agreed upon in order for users to be able to input ogonek-accented characters, in the same spirit as the already existing accent commands; nowadays, it's `\k` in LATEX.[7]

Over the years, more characters were designed and entire alphabets were digitized using METAFONT, starting with Greek and Cyrillic, which were drawn by various people around the world.

An important step was when TEX was extended in 1989 to handle 8-bit input (then becoming TEX3), thus enabling fonts to have up to 256 characters. The next year, during a meeting in Cork, TEX users from all over the world agreed on a standard encoding for TEX's Latin fonts, which then came to bear the name of its birth place (or the alternative, less poetical names of `T1` and `8t`). Another important milestone at that time was the advent of the LATEX babel package, which attempted to provide a convenient way to switch between languages and a common interface for LATEX users.

But even after those fonts were designed, after those standards were agreed upon, many things were left to do: what about Arabic, for example? TEX offered amazing possibilities, but did not really address the issues of right-to-left typesetting and it also completely left aside the fact that characters can have different forms according to their place in a word (both being essential features of Arabic). Therefore, to go further it was necessary to *think different*![8]

2 Think different

2.1 TEX encompasses the *Mare Nostrum*

As early as 1987, the first experiments were made in handling the challenge of Arabic typesetting and gave birth to a modified version of TEX called TEX-XET, to emphasize the fact that it could write in two different directions.[9] This worked in a particular way: when writing data in the output file, TEX-XET did not reverse the order of letters but wrote a mark whenever it encountered a sequence of Arabic letters, and then let all the work be done by the printer driver. That way, the text was stored in *natural* order in the output file — that is in the order in which an Arabic speaker would speak out the letters — but, on the other hand, it meant the files output by TEX-XET had to be processed by a special driver. So 高德纳 — who, once again, was the lead in that project — decided that the output format

[5] Including, of course, all the languages of the former colonizers like English and French, but also important African languages like Swahili which are written entirely in the Latin alphabet.

[6] The Poles are very proud of their *ogonek* and you should not upset them by speaking ill of it. Maybe it is even too daring in the eyes of some to state that *ogonek* resembles a reversed cedilla!

[7] For a thrilling account of how TEX came to Poland, I highly recommend reading the text of this talk, given at the TUG meeting in Hawai'i in 2003, and published in *TUGboat*, volume 24, number 1: `http://www.tug.org/TUGboat/Articles/tb24-1/odyniec.pdf`.

[8] To quote an old slogan of one of the big computer manufacturers — I'm not sure what the legal status of such commercial slogans is and I may not be entitled to reuse it in a document; but I want to make sure people know I didn't mean any harm and in case TUG is sued I deny everything.

[9] The founding article was published in *TUGboat*, volume 8, number 1: `http://www.tug.org/TUGboat/Articles/tb08-1/tb17knutmix.pdf` and makes fascinating reading even today, especially when compared with the current paradigm established by Unicode in that area — the so-called bidirectional algorithm.

input: `Book is "ﻙﺕﺍﺏ" in Arabic`
output: Book is “كتاب” in Arabic

Figure 3: The challenge of Arabic writing: when setting an Arabic text, the order of the letters does not only need to be reversed, but their shapes also may vary a lot — can you recognize all four of them on the second line?

should be called DVI-IVD, to differentiate it from the traditional DVI output format.

In this way, both the TEX program and the DVI format were “extended” in the sense that they were made able to handle different types of information in addition to the ones they already knew how to process or store. We shall meet a lot of these along the way, and we shall refer to them as “extensions” — or sometimes “engines” for TEX extensions. So TEX-XET was, probably, the very first TEX extension.

A few years later, TEX-XET was itself extended via something that achieved roughly the same goals, but without needing to resort to an extension of the DVI: it readily reversed the order of each letter in the output file as appropriate. To mark both the similarities and the difference of this second extension with the first one, it was called by the same but a second hyphen: TEX--XET!

These improvements were interesting and made Arabic typesetting with TEX possible quite early; but it was still an experimental system, and apart from that, it did not change things for other scripts such as, in particular, the Indic and South Asian scripts.

2.2 Enter Unicode

For better-suited treatment of such complex scripts, Omega[10] was designed. It consisted of several major improvements:

- It enabled (probably) every sort of writing directions.
- It came with a set of filters (the Ω transformation processes, ΩTP for short) that transformed the input text.
- It enhanced the traditional font formats used by TEX from 8-bit-based encoding to 16 bits.

The two first points made the treatment of Arabic much more natural (just switch the writing direction from left-to-right to right-to-left, top-to-bottom; and filter the input text to give each letter its appropriate appearance given the context); and the third one was also very important because it addressed the problem which we haven't yet mentioned: up to then, TEX handled only fonts with at most 256 slots.

[10] We shall call it simply by the Greek letter from now on.

This isn't so important for alphabetic scripts, but becomes a major issue when one wanted to typeset in a language using ideographs, whose number by far exceeds this limit.

Ω addressed part of this problem by making direct use of (possibly) very large font metrics; that is, it could use any font on the input but remained constrained by the output format.

Anyway, it brought with it a conceptual leap, even if it failed to address some of the issues of the output format. Over the years it has been successfully used to typeset the Devanāgari, Malayalam, Tibetan, Inuktitut and Cherokee scripts among others, although it has never really gained a wide acceptance.

2.3 The other way: Generating PDF

Ω was first formally released in 1994, and by that time there was a document format that was increasingly gaining in popularity and commercial strength: PDF. Hearing about this “Portable Document Format” in the TEX world, one cannot help thinking that it is a concept quite close to the traditional output format, DVI (does it not stand for “DeVice-Independent”?); therefore it seemed only right that TEX should be able to produce PDF directly: and so it did, with the birth of the well-known pdfTEX on March 15th, 1997 (then under the name `tex2pdf`).

Another huge improvement pdfTEX brought was the direct handling of TrueType fonts, which by that time had become a major font format for personal computers.

2.4 One more εxtension ...

Worth mentioning here, since its later development was to be closely related to pdfTEX's, is “the” extension of TEX, called ε-TEX for “extended TEX”. Developed during the late 90s, it extended the above-mentioned TEX--XET (the second one, with two hyphens) and was therefore of great use to Arabists and other communities writing from right to left.

Some time later, its very useful extended features were merged into pdfTEX, which thus had for a while an offspring called pdfε-TEX, now fully incorporated into pdfTEX; that is, pdfTEX now supports the ε-TEX extensions, but it can also pretends to know nothing about these and be simply pdfTEX.[11]

2.5 Needless To Say

Before proceeding to the last part of this account, there are a few words to be said on another attempt

[11] Just as it also could, from the very beginning, behave as the DVI-producing TEX, or the actual pdfTEX — which means that in DVI mode and *with* the extensions ... you probably get the picture.

of extending TEX, which isn't very famous now but whose name still lingers in many memories. It was to be a completely new concept, opening up a world of possibilities ... but as of today, it is no more.

The "New Typesetting System", as it was called — or $\mathcal{NTS}$ for short — was a complete reimplementation of TEX in Java, aiming at full compatibility with the original engine, and providing at the same time the great modularity and extensibility that comes with that language.

Sadly, while the first goal was actually achieved (TEX was indeed rewritten in Java), it proved completely unusable and pointless because of its extreme slowness. The extension projects were never carried out and $\mathcal{NTS}$ has now been officially declared dead.

3 Rule, TEXannia, TEXannia, rule the waves

3.1 Taming the multilingual lion

Back to living projects now: there is one very young lion which has brought many changes for a lot of users recently. XƎTEX, as it is called, is an extremely multilingual extension of TEX; the very name suggests, again, that it can typeset in every direction (it can spell "TEX" backward). Its spirit is a bit special in that it started off (in April 2004) as a Mac OS-specific program which made heavy use of the Apple libraries designed to handle text and scripts.[12]

Shortly thereafter (April 29th, 2006), XƎTEX was released for Linux too, and it was not long before it was ported to Windows as well.

XƎTEX's main distinction — and its overwhelming advantage for many of its adherents — is to get rid of nearly all the hassle in font selection, font installation, etc., while opening up at the same time a whole new world of possibilities: people can suddenly use the bleeding-edge features of the newest font technologies with no particular problem. The key to this was the use of a lot of external libraries — which of course comes at a price: users lose part of the control over every detail of the processing chain which had always been a great advantage of TEX; but many find this tradeoff acceptable.

3.2 Towards the infinite and beyond

Another TEX engine worth mentioning in passing is called Aleph (ℵ), of Ω-ish ascent. It started as an attempt to stabilize Ω while merging the extensions by ε-TEX at the same time (hence its original name, ε-Ω).

[12] Before XƎTEX there was TEXGX (on the Mac only) which used the same series of Apple libraries, then called "GX technology" — TrueType GX was an extension of the TrueType font format, now replaced and enhanced by AAT — Apple Advanced Typography.

While it attracted much attention for a few years after it was launched in 2001, it is today overshadowed by another successor to TEX, which is now thought of as representing the future path.

4 Howling to the moon

This "successor" is LuaTEX. As this seems to be yet another prefixed version of TEX, we shall first explain what that prefix is. Lua is a small scripting language originated at a university in Rio de Janeiro (Brazil) which was developed to be embedded in other applications. The word "lua" means moon in Portuguese (hence the title of this section).

So the idea seems clear, LuaTEX is Lua + TEX: an embedded language in TEX, enabling us to go even further than anything that could be done before with macros; in LuaTEX we will also have the Lua language and we can write Lua functions in addition to TEX macros. And ... there is actually much more: while LuaTEX was indeed meant to be Lua + TEX (actually pdfTEX, now merged into pdfε-TEX) when it was first conceived in the beginning of 2005, it is now also incorporating the features of ℵ and its parent Ω, therefore effectively merging two families of engines: the "Ω way" and the "pdfTEX way". Lua will be present at every stage of the processing chain, with *callbacks* enabling the user to redefine parts of TEX's tasks using Lua functions. Finally, METAPOST is planned to be part of it too, being rewritten as a library (instead of a stand-alone program).

LuaTEX is under active development today and a first public release is planned for the 2008 TUG conference in Ireland.[13]

5 Back to the future

With LuaTEX we have reached the most recent developments in TEX, and here it seems nice to say some words to summarize the changes that we have seen above.

If any general view is to be had, it seems to me that the main changes that TEX has undergone over the years were not only major improvements but genuine *Copernican Revolutions* which progressively widened TEX's field of application. I have tried to classify those phases in the article by making each one of them a different section: section 1 shows how TEX started with an approach of typesetting akin to that of the craftsman's carefully setting type to build a page of text,[14] while undergoing an initial modest expansion along with some "standardization" (Babel

[13] When TUG will return to Cork, which became famous in the TEX world 18 years ago!

[14] Let us not forget: "Rhymes are typeset with boxes and glue", in *The TEXbook*, chapter 14.

package for LaTeX, Cork encoding). Then it went through a phase where the first experiments were made to handle "complex scripts" (section 2) and this gave birth to the first true extensions of TeX which are actually quite old (again, TeX--XeT was written 20 years ago). These extensions were consolidated in the recent past described in section 3, when TeX showed how it could still keep up with major changes in the printing industry (PDF, TrueType and then OpenType fonts). The present of TeX development (section 4) is exemplified by LuaTeX which, once again, comes with a complete change of perspective on TeX processing. These have been the four "ages" of TeX.

Writing history is important, and I have always got the impression that the TeX community did not care enough about its own history — there are of course well-known bits and anecdotes about TeX, but those are closer to legend than to history. Writing history is also a difficult and humble task, and I cannot claim having covered everything that was worthy remembering about TeX: some words could have been said about macro packages (beyond LaTeX, which I simply quoted in passing) as well as TeX distributions which have contributed a lot to TeX's expansion. Nor have I talked about important "industry standards" such as XML which have also become an important part of TeX's capability today (this would have been more linked to the macro packages than to the engines themselves). It is therefore my hope that we can, maybe, enhance this article with more descriptions and memories, and I have opened a small page at the ConTeXt wiki to discuss this: `http://wiki.contextgarden.net/History_of_TeX`.

To conclude, I would like to name a few places where I've personally encountered TeX, as it gives an idea of the versatility of TeX and the extent of the *Pax TeXnica*:

- The general scientific community and especially mathematicians and computer scientists.[15]
- People from humanities, especially in Ancient Greek and linguistics.[16]

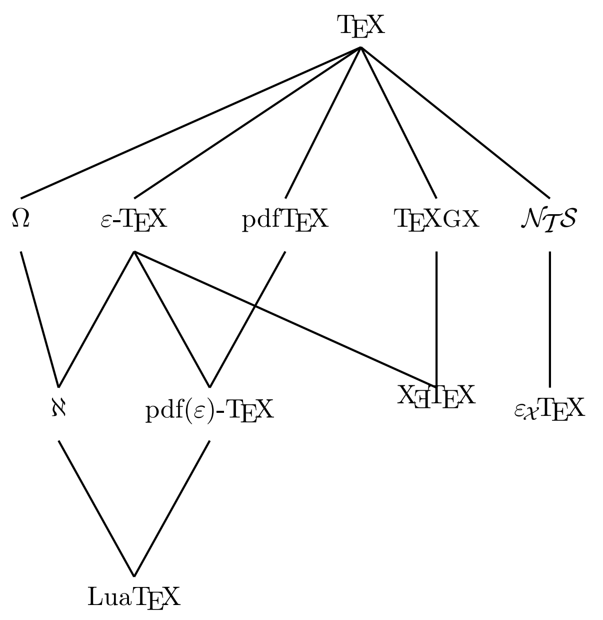

Figure 4: The happy TeX family. The different extensions have been divided into successive "generations" of engine, corresponding to the different sections in this article.

- Shopkeeper in one of the biggest Chinese bookshops in Paris.[17]
- Musicians needing to engrave scores.[18]
- Users of free software.[19]
- People involved in the publishing industry.[20]

[15] This was obvious but I felt I still had to mention it first!

[16] Indeed, what other free program can handle at the same time Ancient Greek, Russian, Lithuanian, Latvian, Sanskrit and French? Someone at my university was doing a Master on Indo-European linguistics and did really need to input all these languages.

[17] Perhaps my most amazing encounter with TeX in a place I didn't expect it at all, but I swear it is true: while gazing at the shelves of the aforementioned bookshop I overheard two members of the staff discussing how to produce documents in Chinese (probably for the shop's catalog).

[18] MusicTeX and MusixTeX have many an adept in spite of their extreme difficulty to master.

[19] Indeed, on an average Linux distribution, there are very little software able to rival TeX — OpenOffice is an obvious example, but it may be the only other one.

[20] Especially for processing XML, as already mentioned.