Interview with Javier Bezos

Paulo Ney de Souza

Editor's note: This interview took place on 25 July 2020, during the TUG 2020 online conference.

Paulo Ney de Souza: Nice to meet you, Javier. Your first time, and half in person.

Javier Bezos: You're welcome.

PN: First let me introduce Javier to the people here that may not know him. 10 years ago he has assumed maintenance of one of the most basic and most famous packages of LaTeX, which is Babel, and is also someone deeply interested in issues of production and use of TeX in production, and so forth.

Welcome to TUG 2020. Let me see if I can start shooting you some questions.

First thing I'd like to hear is your background. How did you grow up, how did you get here? Can you tell us a little bit about your background, independent of TeX?

JB: Well, I was born in the Canary Islands, although I spent most of my life in Madrid. I think there is nothing special to tell, or it doesn't seem so to me.

My liking for language issues and the world of books, go back to my high school days because I was one of the founders of the school paper, and I was passionate about having a nice look with all the letters in their places, are the types right, ...

Then my parents asked me if I'd like a Spectrum.[1] And this changed my life in many ways, because I discovered something for which I had a natural intuition. I wasn't interested in games. They quickly bore me, and I began to program. I programmed a little word processor, and then I started to write in assembly language. When I remember now how much my parents spent in books on the Spectrum—; they were difficult to find, and very expensive. I'm very grateful to them for supporting me in almost anything when I wanted to dedicate myself to something.

When I had to decide what to study, the first thing I thought was physics, physical science. But I suddenly changed my mind. And finally, although it seems a bit strange, I decided to study radio, and that's what I did for seven years in a classical music radio station.

But even then, I didn't stop my TeX jobs and other subjects and actually right now my main activity is as linguistic advisor, which has little to do with the radio, but I am a person with very broad tastes.

PN: Can you tell us how did you meet TeX for the first time?

JB: Thanks to my wife's thesis. Wife now; she was my girlfriend then. Before, I already worked with Ventura, an incredible program, and I must confess that at first, LaTeX seemed incomprehensible to me.

I got the Knuth book and it looked to me like a mess. But I could cope with the book by Kopka and Daly, and there I began to understand what it's all about.

In Spain, there was no TeX group, so I signed up for DANTE. So in addition to learning more about TeX, I practiced a little German.

I already said I like programming. So I started writing small macros, then not *so* small macros, and finally packages. I usually say that there are those who solve Sudoku to entertain themselves, and I program. Especially, but not exclusively, with TeX.

PN: Interesting, interesting. What changes have you seen in the document production world and in the TeX world during your career, since this time that you began, all the way to now?

[1] An early UK home computer from Sinclair.

JB: The love for a well-made book at all levels, doesn't seem as popular as it was years ago. The phrase "Go forth now and create masterpieces of digital typography" doesn't make a big impression today. I also notice there is a great tendency to believe that typographic skills are built on the handling of a program when that's just the starting point. The point is, design is regarded more as the creative aspect than the practical one for the needs of actual documents.

But let's be fair. We can't say the love for a well-made book was universal formerly, either.

PN: Aha. I had an interesting discussion with Boris yesterday, and it seems that TEX, what with the size of the user base, TEX is one of the world's oldest subsystems. And that is 1977 to now, quite a few years. Do you think that it will continue forever?

JB: I think now there may be a little revolution going on with the arrival of LuaTEX. It has its loose ends, but it's an important advance, and its possibilities remain to be seen.

In fact, I'm using it a lot in Babel, because it has allowed me to introduce functions for non-standard hyphenation and for transliterations. Not to mention bidi writing and line breaking where we can have almost absolute control.

PN: What packages, what kind of tools do you find essential to your work right now as a document author, as well as a producer; what do you do use on a daily basis?

JB: I don't think that there's a simple answer for this. When I start reading LATEX package manuals to see what they are all about, I'm still amazed at what they can do. And sometimes I wonder how I've been able to live without a certain package.

The packages I use the most are the standard ones: `hyperref`, `graphicx`, my own packages, and also a package by Zdeněk Wagner for the page layout (`zwpagelayout`).

PN: And so when you're producing a book do you use exactly the same thing as an author?

JB: No, it depends on the project...

PN: And what do you use mostly with that flow, to produce your own work? Do you use LuaTEX mostly for producing your own work?

JB: I usually write my own macros. So I like to reinvent the wheel.

PN: [laughter] Let me ask you this. You are a guy that knows a lot about other issues, especially things like typography and orthotypography. Do you think it's worth spending a lot of time on certain issues in typography, even though very few people notice?

JB: Oh, sure, they notice it, although without realizing it. I want to look at books which invite reading.

The problem is there are publishers who believe the opposite, and not only in the field of typesetting, but also, for example, in copyediting. I don't know how the situation is in other countries, but in Spain, many media have dispensed with copyediting.

PN: Aha, so the production is substandard, as we evolve into the 21st century.

JB: Yeah.

PN: Do you use other tools besides TEX in your work?

JB: In my day job, no, technically no. As I'm mainly a linguistic advisor, I use it for complementary tasks. I work at a foundation called Fundéu, directly linked to the Efe news agency, and partly to the Spanish Royal Academy, whose function is to help the media in their use of Spanish.

But there can be a connection, because writing and pronunciation of names from other languages is another of the tasks of the foundation, and I'm something like the specialist in romanizations. So I can see there is a connection, even if it's more between my daily work and what I'm doing in Babel.

PN: You do a lot of work on a certain area where not very many people understand exactly what it is, which is orthotypography. How did that come about, how did you become interested in such a specific area of typography? But first, would you explain to us what orthotypography is?

JB: Basically, it is not very different from a combination of orthography, style and typography. I'm not sure exactly what orthotypography is, but well, in Spain it is a concept you should use very often. And if I'm right, in French too.

PN: On this point of orthotypography, I want to bring an issue which is very dear to mathematicians.

There are two camps in mathematics in terms of punctuation and in formulas which are central to a page. There's the people that believe that it *should* be punctuated, and there's the people that believe it should *not*, that punctuation is to use a pause, and that, and that if you have a formula which is centered on the page, that pause already exists.

What is your idea on that?

JB: If you put it when the formula is in line, at the end of a sentence, why shouldn't we put it when displayed?

Anyway, in typography, what matters is to have good taste. Good taste and good eye. Then we can have your rules of style. But if they don't work for us in a certain context, you shouldn't hesitate to think of other possibilities. After all, it's still an art, like painting, like composing music. Music may follow many rules when we are learning, but the masterworks are being created by breaking the rules.

PN: Interesting. Very interesting.

Let's talk about Babel a little bit. I mean about 10 years ago, a little bit more than 10 years I think, you took over as the main lead maintainer of one of the most used packages. How has it been? Tell us.

JB: My interest in the localization issues came from much earlier. Already in 1997 I was considering creating an alternative to Babel, which I called Polyglot, but it wasn't long before I realized it was a futile effort because Babel, after all, mostly did the job, and it was the standard.

Then came another opportunity, Omega, by Yannis Haralambous and John Plaice, and I started working on it. This time, the package was named Mem, a Hebrew letter, but we know that Omega sadly didn't develop its full potential. So Mem also failed.

Then Frank Mittelbach proposed to me to join the LaTeX team (I think that was in Brest, EuroTeX 2003), which I gladly accepted because that way I could help in this area, and possibly others as well. And I still continue.

PN: How do you do your work on Babel? How do you connect to authors, languages, and the production of the `.ldf` files ... could you tell us a little bit more about that workflow? How do you find the people to help out with the languages?

JB: In many cases, I generally don't like interfering with the work of those who develop national styles, some of which are supported by their respective user groups. But I'm always willing to help and collaborate. Also, if I discover there is someone interested in supporting some language, I contact him, as I did for example with the group for Kurdish.

On the other hand, I like to investigate by myself. I'm self-taught, self-taught by nature. And I like to learn things by myself. I think I'm good at it and I really enjoy doing it.

Sometimes it means you end up reinventing the wheel. Sometimes that wheel is worse than the others. Other times it turns out to be better. Of course, that doesn't mean I don't like suggestions, help or whatever. Quite the opposite.

PN: I was doing some research a few weeks ago on Babel and other things that you have written in order to understand your work a little bit more, and I came across with a package that you maintain for the language of Guaraní. This is my mother's tongue. Literally. My mother was a speaker...

I was extremely surprised to see this. This language is spoken by a lot of people, but without formal production of books. It's a language spoken in South America, from the north of the Amazon, all the way to the south, by indigenous people of South America.

How did you connect with this language?

JB: So many years ago. I just don't remember why I did it. It's true that for a time, I was interested in the languages of the Spanish-speaking countries, but I don't remember more details, unfortunately.

PN: Ok. I guess one question that a lot of people are interested in is about HarfBuzz, how LuaHarfBuzz is changing, and may change Babel.

JB: Very little, with relation to the original LuaTeX which I have been working on for some years.

Mainly because HarfBuzz is about fonts, not about languages, and sure, the fonts contain the information about how to render a script, but I think HarfBuzz is not exactly part of Babel. More of the LaTeX kernel.

There is a possibility to make HarfBuzz the default shaper, but as Marcel [Krüger] explained, I think it's not a good idea. Because there is a problem with how HarfBuzz deals with hyphenation points in the Latin, Cyrillic and Greek scripts, because many hyphenation points are lost. It depends on the font, but in those scripts in most cases the best option is to stick to the default renderer.

PN: Where do you see the future of Babel?

JB: Serving as a framework to localize and internationalize documents at several levels. Until now, Babel has followed a model which I call *vertical*, based on language styles which are basically self-contained black boxes. That's fine in monolingual documents, and in fact is and will continue being the recommended method in these cases.

But when we need to combine several languages, we can have serious internal relationship problems. So I'm working on a new model, which I call *horizontal*, where languages can be defined in a descriptive way, with the help of `.ini` files based on key–value pairs.

That allows us to reuse a solution for one script in another without too much hassle. For example, traditional line breaking in Amharic turns out to be almost identical to that of a southeast Asian script. So it was enough to adjust the keys to get the right result.

PN: I have a specific question that comes from the mathematicians' camp... will there be a day in which we will be able to choose mathematical functions based on their own languages?

JB: It's something that has been requested sometimes, but on the other hand, it's a controversial issue — should they really be translated? And after all, it can be solved easily with `\renewcommand`, so I wonder if it's something necessary in Babel. Well, I think it should not be ruled out beforehand, but it's not one of my priorities.

PN: The biggest complaint I hear, Javier, is that if you have a big shoe, your foot grows, and then you start seeing books in Portuguese with "sine" instead of "seno".

JB: The same in Spanish, yes. And there are various others written with an acute accent, traditionally.

PN: So let me ask you about some modern questions. One of them is about Polyglossia. Polyglossia was sort of started right around the time that you became a development leader on Babel, and Polyglossia got started as a replacement for Babel for XeTeX and LuaTeX. Then the development of Babel picked back up again with your work. Do you see some synergy happening between the two packages?

JB: Well, actually Polyglossia is older, about five years I think, and on the XeTeX list some people even suggested I take charge of it, but I did not for several reasons. One of them was my personal situation at the moment, but another was my experience with Polyglot, which I mentioned earlier.

By the way, if you wonder if these names are related, the answer is yes, they are. There was discussion about a multilingual package for XeTeX and I proposed this name because I wasn't going to use it any more. And that name, but in Greek instead of English, stayed. I don't remember who proposed the translated name, but it was a good idea.

When I created all the new `.ini` files, I did it thinking they could be useful for both Babel and Polyglossia as a shared resource, and not only for them, but also for other packages.

Well, this work was done and now we need to fill it, which I'm doing gradually. There are already about 200 more-or-less completed languages or easily completable, and there are also templates for about 500 languages, and they can be a starting point.

PN: So I see that you are, you seem to be fully BCP 47 compliant, at least on the naming of the files and so forth.

JB: Yes, sure, we can already deal with BCP 47. What's more, you don't even need to declare languages in the preamble, because Babel can load the locales on the fly, with some limitations.

Thinking about use cases, I realized there are texts generated externally, and the main document may not have the slightest clue of the languages needed. An example can be a bibliography, with many names and titles from a multitude of languages.

PN: Let me ask you a question now about preproduction and production using PDF. The open source community tools seem to be completely unprepared for dealing with prepress. We have ways to check PDFs, we have problems with resolution, colorspace, flattening, transparency, overpainting, and I see that you started doing some embryonic work on the package `colorspace`. Can you tell us where were you going then?

JB: Yes, this is a problem. I'm sure the work in progress will bear fruit. For example, Joseph Wright and Ulrike Fischer are working on improving the color support in LaTeX, so that what `colorspace` does will be better done and avoiding the conflicts between different packages trying to generate PDF code directly.

On the other hand, we should never think a document created with, say, InDesign is automatically ready for printing. I myself have suffered from poorly created documents by designers who think it's all about creating frames and things like that. They may not even know how to correctly import a PDF image.

Here again, I think that LuaTeX could be of great help. For example, I think it should be possible to write a utility to convert calibrated colorspaces. I'm not sure, but I think ConTeXt can do it.

PN: That's great news. LuaTeX can be used for reproduction. So do you plan to take `colorspace` into a full color production package for PDF?

JB: For the moment, my work on `colorspace` has stopped. There is no point in continuing with it if its features will be supported directly in the LaTeX kernel.

PN: Thank you. We have a question from Michael Topp, on the net. What compilation engine do you generally use or recommend?

JB: I use LuaTeX since many years ago.

PN: Thank you very much, Javier. I really appreciate your coming for this interview with us.

JB: Thank you.

◇ Paulo Ney de Souza
`tug.org/tug2020`

A Addendum: Recent advances in `babel`

Javier Bezos

The purpose of this addendum is to highlight some of the progresses made in `babel` for it to become, as explained in the interview, a localization framework. New features are announced and explained in some detail in the `babel` wiki, `github.com/latex3/babel/wiki`.

No program can be considered finished. The world continues to evolve, and there will be areas deserving improvements. So, expect new features in the future. For example, at the time of this writing, there is work in progress to deal in a coordinate way with chapter counters (Hungarian, CJK), lists and other labels.

Locale data. There is a large database with `.ini` files directly usable by `babel`, based in data taken from several sources, including the Unicode Common Language Data Repository.

I think these files will help to solve what I think is one of the most common problems when creating a language style based on TeX code, namely, there are linguistic communities without a TeXnician at hand to develop it. There are templates for about 500 languages in the GitHub repository, which can be used as a starting point.

One of the main advantages of the `.ini` files is that the data is provided in a descriptive way. For example, the file for German (`babel-de.ini`), contains things like:

```
[identification]
...
name.local = Deutsch
name.english = German
name.babel = german
name.polyglossia = german
tag.bcp47 = de
language.tag.bcp47 = de
tag.bcp47.likely = de-Latn-DE
tag.opentype = DEU
script.name = Latin
script.tag.bcp47 = Latn
script.tag.opentype = latn
...
[captions]
preface = Vorwort
ref = Literatur
abstract = Zusammenfassung
bib = Literaturverzeichnis
...
[date.gregorian]
date.long = [d].[ ][MMMM] [y]
date.short = [dd].[MM].[yy]
months.wide.1 = Januar
months.wide.2 = Februar
...
```

I invite all linguistic communities to complete them, and to make suggestions and feature requests for any language to improve the LaTeX support in the three main engines. Feel free to fork the GitHub repository and to make pull requests.

BCP 47. BCP 47 tags can be optionally used to select languages, but note since a two- or three-letter word can be a legitimate language name, BCP 47 codes are not activated by default. In fact, the BCP 47 tags have been defined in `babel` for years, but they were not easily accessible nor usable. In the most recent versions, you can select a language (its most basic features) merely by writing something like `\foreignlanguage{zh-Hans}{...}`.

I wouldn't like to go too fast in a more or less complete implementation of the subtleties of BCP 47 tags. I prefer to go step by step, much in the spirit of continuous development, but now the most basic tags are already there.

Line breaking and bidi writing. Thanks to LuaTeX, `babel` supports bidi writing without explicit markup and the three basic methods for line breaking: CJK, southeast Asian and Western. You can even change the properties related to them for specific characters; for example, the direction may be changed from 'Arabic letter' to 'Other neutral'.

Automatic font and language selection. Sometimes, you need to insert short pieces of text in a different script from the main one; for example, some Bangla words inside Bulgarian. Now, language identifiers (`\language` and `\localeid`, which currently apply mainly to line breaking) and fonts can be automatically set based on script blocks, without explicit markup.

Transformations. Another feature related to LuaTeX, still under development, although the basic features are already usable, are the commands `\babelprehyphenation` & `\babelposthyphenation`, which allow transliterations and non-standard hyphenation. For example, with the following settings "ff" is automatically divided as "ff-f", and also "mm", "tt", "rr", "pp":

```
\babelposthyphenation
  {german}{([fmtrp]) | {1}}
  {
    { no = {1}, pre = {1}{1}- },
    remove,
    {}
  }
```

Counters. They can be defined in `ini` files by enumerating the elements:

```
lower = a b c d e f g h i l m n o p q r s
```

The same applies to additive numerals; examples are Greek, Syriac and Japanese.

◇ Javier Bezos
jbezosl (at) gmail dot com