

## What's in a name?

*Hans Hagen*

### 1. Introduction

I sometimes wonder how much the fact that English is the language mostly used in programming environments influences the way one looks at a program. For instance, translating the names of an operating system *Windows*, an image manipulation program *Photoshop* or a text editing program *WordPerfect* to Dutch makes them sound kind of silly to me. The name can influence what you buy or are willing to use. These are examples of commercial programs but there are plenty examples of such naming in the open source universe too. I write this in my own bad English so that other non-English speakers can try to do a similar exercise.

So, when I was reading an article about cpu technology called *thread-ripper* and after a while also saw the usual talk of yet another bunch of technologies marked as 'stack' and translated that to Dutch it again made me feel somewhat puzzled about such names. From there it was a small step to wondering about programming languages, and especially the ones I use: T<sub>E</sub>X, MetaPost, and Lua.

One can even wonder to what extent the quality of programming is influenced by the names of commands and keywords. A language named *Basic* sounds less serious than *C*. A meaningless *Lua* sounds different than *Python*. Does using your native tongue make a difference? In Dutch and German words tend to get long. When I look at my French dictionary it is rather thin, but we might need accented characters. Words in a language like Polish can differ per usage. What if German or Spanish had been chosen as the language for what is now the United States? How would we perceive programming and what would look natural to us?

### 2. T<sub>E</sub>X

The T<sub>E</sub>X language comes with a lot of so-called primitives built in. Many of these relate to concepts in the program. For instance, a movement in horizontal or vertical direction that can stretch or shrink depending on what the boundary conditions demand, is called 'glue'. When discussing this in Dutch the word 'lijm' can be used and after seeing it a few times it might sound ok. We can probably use 'elastiek' ('elastic').

This internal concept is actually represented to the user via the interface name 'skip', take: `\abovedisplayskip` and `\belowdisplayskip`. Here the word 'display' refers to math that gets vertical space around it and is normally typeset in a somewhat larger way compared to 'inline'. The word 'skip' can be translated to 'sprong' (translated back we could as well get 'jump'). But how to translate 'display'? An internet translation can be 'tentoonspreiding' but apart from it being a long word it sounds pretty weird for something mathematical. The combined translation of such a command will not work well so I think we probably need to create completely different words to describe these quantities. Taco suggested that `\bovenuitstallingkortesprong` might work for `\abovedisplayshortskip` but luckily no ordinary T<sub>E</sub>X user will set such parameters in a document source.

In ConT<sub>E</sub>Xt we use the somewhat typographical term 'wit' or 'witruimte' for vertical spacing. Some parameters like `\baselineskip` can be translated directly to the Dutch `\regelafstand` which is a proper typographical term (T<sub>E</sub>X has no concept of line height). Okay, it can become messy when we translate `\lineskip` by `\interlinespace` as that could be seen

as the baseline skip too ('interlinie' comes to mind). Quite a mess. In many cases we probably would not handle the skip part in a parameter: `\leftskip` could become `\linkermarge` and `\parfillskip` could become `\paragraafuitvulling`.

Another concept is that of 'penalty', or in Dutch 'boete'. It's probably harder to get the combinations right, simply because they have no typographical meanings, they're more process controllers. I fear that most translations would sound pretty weird to me. So, how do they sound to a native English speaker? Words like 'club' or 'widow' can be translated to their Dutch gender neutral counterparts 'wees' and 'weduw' but how strange does `\weduwboete` sound?



The counter variables are easier. When they end on 'char' that can become 'karakter'. However, translating `\escapechar` with `\ontsnappingskarakter` might look a bit weird, but as that one is used very seldomly, a weird one doesn't matter much. Operators like `\advance` and `\multiply` could become `\verhoog` and `\vermenigvuldig` which doesn't sound that strange in this context.

There are 'rule's and 'box'es. The first one could be translated to 'lijn' which sounds quite good. But what to do with the second one. We can use 'blok' (which translates back to 'block') which is good when we start stacking things, but also with 'doos' which is more literal but sounds to me somewhat silly: `\hdoos{whatever}`. I'm not so sure if I would have seen that in a book about  $\text{\TeX}$ , I'd looked further into the language. The optional keywords 'width' etc. could be translated well into 'breedte' etc., so no problem there.

There are all kinds of very peculiar aspects that need a translation. For instance the (for new users intimidating) primitive `\futurelet`. The 'future' part is no problem as 'toekomst' isn't that weird but the 'let' will for sure become something very long in Dutch, so we end up with `\toekomstigetoekenning`, but seeing that long one, we can consider `\kijkvooruit` as reasonable alternative. It definitely leads to more verbose programming.

Expansion is a tricky one. I have no clue what would make nice translations of the primitives `\noexpand` and `\expandafter`. The Dutch 'uitbreiden' simply is not sounding good here. Taco Hoekwater came up with a good alternative 'uitvouwen' for 'expand' and I like that one because we then can let bookmaker (a somewhat dubious term in itself) Willi Egger organize a workshop in unfolding (instead of folding).

Talking of 'macros' is less a problem because there is no Dutch word for it. There are more words with no real translations: `\kern` for instance probably would need some thinking but there might be a typographical equivalent that could be used.

The  $\epsilon\text{-TeX}$  and  $\text{LuaTeX}$  extensions introduce new names, like `\detokenize`, `\boundary` and `\attribute`. The first one is hard to translate because again it relates to an internal concept: 'tokens'. I get the feeling that translating each occurrence of 'token' by 'teken' kind of makes everything look less serious. To strip something

from its special meaning, which is actually what `\detokenize` can give weird translations: `\onttekenen` is not really a Dutch word so a completely different one has to be found that describes what happens, like `\ontwaarderen`. On the other hand, `\boundary` and `\attribute` can translate directly into `\grens` and `\attribuut` where the last one sounds mostly okay.

Just to get you thinking: how would you translate `\looseness` ('losheid', related to linebreaking), `\deadcycles` (`\zinlozelus`, in the perspective of building pages), `\pretolerance` (again line break related, here we can use something `\tolerantie`) and `\prevgraf` (which is actually even in English a weird one but hardly used anyway, so Taco likes `\voorloopregels`)? The easy ones are `\omit`, `\meaning`, `\number`, maybe even `\mark`. The for users often difficult to grasp 'cat-code' can be simplified to 'code' which is proper Dutch. Concepts like 'align' translate well to 'uitlijnen'. Short ones like `\wd` could be a problem but any two letter combination can look bad, so `\br` could do. In the same fashion `\def` is ok as it is also the start of the Dutch 'definitie'. Mathematical terms like 'text', 'script' and 'scriptscript' can be confusing: 'tekst' will do but 'schrift' is strange.

Conditionals are not the hardest part: `\if` becomes `\als`, `\else` becomes `\anders` and `\or` is `\of`. However, turning `\ifcase` into `\inhetgevaldat` can be over the top. The `\every...` register variables can also be translated quite well, by using the `\elk` or `\elke` prefix. They are seldom seen at the user level, so no real problem there.

The 'group' related commands are easy as 'groep' is a good Dutch equivalent. Even 'global' operations translate well (`globaal`). A dubious one is `\font` because we can use `\lettertype` but it's not really a translation. The internet translations tend towards 'fountain' kind of things.

The concept of 'discretionary' again needs a decent typographical translation although

`\hyphenation` can become `\afbreking`, translating `\discretionary` needs some imagination. The concept of 'leaders' is again something that can best be bound to something more typographical because `\leaders` turned `\leidinggevende` is not an option nor is `\leiders`.

The prefix `\un` as used in `\unhbox` can become `\ont` so that we get `\ontdoos` but I get the feeling that this one can be source of jokes. The more verbose `\pakdoosuit` (equivalent to `\unpack-box`) would do better. To translate `\unvcopy` into the gibberish `\ontdoosdecopie` is simply ridiculous and `\copieeruitgepakte` is a bit long. The `\lower` and `\raise` on the other hand translate well to `\verlaag` and `\verhoog`. Keeping `\relax` untranslated sounds ok to me, because `\ontspan` really makes a language silly.

### 3. MetaPost

The TeX language is driving a macro system while Lua is a procedural language. The MetaPost language sits somewhere in between. It is still expanding all along but it looks a bit more like a programming language with its loops, assignments, conditionals, expressions and (sort of) functions. As a consequence some of what I mentioned in the previous section applies here.

Translation of, for instance, `truecorners` into `echtehoeken` can give the language a less serious image. Words like `linejoin`, `linecap` and `miterlimit` relate directly to the PostScript language so translating them also relates to translated PostScript.

The primary, secondary keywords can be nicely translated into serious counterparts `primaire` and `secundaire` which are words that are not really of Dutch origin anyway. The `precontrol` and `postcontrol` words relate to concepts but even there the verbose `controlepuntvoor` and `controlepuntachter` could do. However `punt` as translation for `point` can be confusing because we also use that for `period`. Translat-

ing controls and curl needs some imagination. Words like tension becoming spanning is still acceptable soundwise. However:

```
voor i=1 stap 2 tot 10:
.....
eindvanvoor; % or: eindvoor
```

Kind of interesting is translating `if` into `als` because `fi` then becomes `sla` which is ‘lettuce’ or, when seen as verb, ‘hit’. The `true` and `false` keywords becoming `goed` and `fout` is no problem.

Turning `atleast` into `opzijnminst` at first sight looks strange but actually I can appreciate that one. And `tussendoortje` as translation of `interim`, I can live with that one too as it sounds funny. Concepts like ‘suffixes’ need thinking but `uitdr(ukking)` or more literal `expr(essie)` for `expr(ession)` are okay. The `expandafter`, `scantokens` and similar keywords share the problem with  $\text{\TeX}$  that they relate to concepts that are hard to translate.

The `redpart` and similar keywords could be translated into `rooddeel` but `roodkanaal` (meaning `redchannel`) might be better or maybe `ro-decomponent`. As with  $\text{\TeX}$  grouping related keywords are no problem.

A `pencircle` becomes `pencirkel`, `odd` becomes `oneven`, `reverse` becomes `omgekeerd` (or `andersom` or `tegengesteld`). For `length` we use `lengte`, and so on. All these sound professional enough, just like ‘corner’ related keywords becoming ‘hoek’, although there a clash with ‘angle’ is possible. I’m less sure about `clipped` becoming `afgeknepen` or `begrensd` but `bounded` then needs some thinking as these all are more or less the same. The concept of ‘stroke’ maps onto ‘tekenen’ or ‘vegen’ but lucky for us that one is not really used, contrary to `draw` that can map onto `teken`, while `fill` and `vul` match well too I guess.

The transformations are no problem but I’d use a directive instead: `rotated` or `roteer`, `slanted` or `schuin`, `scaled` or `schaal`, and `transform` or

`transformeer`. As you can see, these have a reasonable word length too.

The concept of a picture is known in Dutch as `plaatje` or `tekening`: not an easy choice. Using `kleur` for `color` is no problem at all. A coordinate pair becomes a `paar`: close enough not to give subjective side effects. The `inner` and `outer` keywords translate well to `binnen` and `buiten` but in code it might look a bit strange.

So, in general, the translated commands are not that weird but still a graphic defined in Dutch keywords instead of English to me might look less serious.

## 4. Lua

We now arrived at a more traditional programming language. The Lua language only has a few keywords. I suppose that it’s just a matter of time before one gets accustomed to `als ... dan ... anders ... eind` instead of `if ... then ... else ... end`. The loops also translate rather well: `zolang ... doe, herhaal .. totdat, voor ... in ... doe ... einde` are all not that verbose. Also, with proper syntax highlighting they stand out and become abstract words. But because examples for kids are normally in Dutch, using a Dutch programming language might give a toy language feeling.

The `local` directive is a bit of a problem because it should be `lokale variabele` in order to sound ok in a sentence. The `goto` should become `ganaar` which is also two words with no space in between. The `function` keyword can become `functie`. A `coroutine` is a challenge (also conceptually); we do have `routine` but how about the `co` part?

Because Lua is such a clean language it doesn’t really end up bad. In C there are some more issues due to the abbreviated `struct`, `int`, `char`, `enum` and `typedef`. A literal translation of `void` to `leegte` to me sounds a bit strange. What to do with `unsigned`? Coming up with some-

thing (short) Dutch for return is not easy either. Translating switch into schakelaar looks like a bad idea but after consulting Taco using keuze came up. The break then can be klaar which roundtrips to ‘finished’ and default can be anders which roundtrips to ‘otherwise’ which is indeed what some languages provide... But, there are programming languages out there that have plenty keywords and that are more challenging. But as I’m a happy Lua user, I don’t have to worry about them.

## 5. Conclusion

Looking at a program source in Dutch the general feeling probably will be different. A low level bit of T<sub>E</sub>X is the worst. For MetaPost it’s bearable and for Lua it is kind of okay. But in all cases, I’m not convinced that it would give me the same feeling. The abstraction of the language due to it not being my native tongue makes a difference. This problem is not much different than what we have with popular music and songs: for non-native speakers it’s basically sounds, but for a native speaker it is more clear when nonsense is sung.

The same can happen to me with movies, where watching some Scandinavian series is different from watching a Dutch one. In the last case one picks up different nuances, not necessarily for the best. But it can be worse: post synchronized (audio) translations can be pretty unbearable and might compare well to programs translated into for instance Dutch. So let’s not discuss the way Germans would deal with this.

## 6. Side notes

We now see monospaced fonts showing up that provide ligatures for e.g. <= and I’ve seen examples where ligatures kicked in for fi. One can wonder about that but ligatures are definitely something to keep in mind when translating.

The ConT<sub>E</sub>Xt macro package is normally used with the English user interface. But the design is such that one can provide different ones too; after all it started out Dutch. It is beyond the scope of this musing to discuss the problems with translating typographical concepts between languages, especially when there are no distinctive words. But it can (and has) been done.