

Mijn leven met T_EX als student

Hoi! Ik ben Dennis Holierhoek en ik ben student aan de Christelijke Hogeschool Windesheim te Zwolle. Onlangs kreeg ik een uitnodiging van Frans Goddijn om eens een stukje te schrijven voor de MAPS over het werken met T_EX als student, en ik voelde mij enorm vereerd! Met toch wel enig gevoel van trots presenteer ik u hier mijn artikel en wens ik u veel leesplezier!

Voor dit artikel zal ik een aantal termen gebruiken uit de wereld van de applicatieontwikkeling en het moderne onderwijs. Ik snap dat niet iedereen met deze termen bekend is, daarom zal ik een woordenlijst bijhouden. Het kan misschien wel zo zijn dat ik op sommige punten iets teveel woorden omschrijf en dingen uitleg die vrijwel iedereen weet, en op andere punten dingen niet weet uit te leggen (voornamelijk de stukken die gaan over ICT) voor iemand die niet uit „mijn” hoek komt.

Ik wil vooraf ook zeggen dat ik al mijn kennis over LaT_EX en vooral over goede documenten schrijven zelf heb opgedaan, en ik dus misschien een paar onhandige fouten maak waar ik mij niet van bewust ben. Wees dus alstublieft niet te kritisch bij het lezen van dit artikel.

1 Ik en T_EX door de jaren heen

1.1 Eerste stapjes

De eerste keer dat ik de term LaT_EX heb horen vallen, was in ongeveer 2012. Ik zat toen in de derde of vierde klas van de havo en was een jaar of 16. Een klasgenoot van mij was enorm fan van Linux-besturingssystemen en stak mij met zijn enthousiasme aan. Windows ging van mijn computer af en ging vrolijk aan de knutsel met Linux.

Tijdens mijn uitpluiswerk over de mogelijkheden van Linux, zag ik veelvuldig LaT_EX langskomen. Ik deed er destijds nog niet zoveel mee, ik zag de noodzaak er niet van in. Maar ik onthield wel de naam.

In 2017 begon ik met een opleiding hbo-ICT op hogeschool Windesheim te Zwolle. Het ging goed, ik haalde goede cijfers en was altijd op zoek naar dé perfecte manier om iets te doen. Zo kwam ik in 2018 op het idee om T_EX te gebruiken. En op 21 mei 2018 om 19.50 was het

een feit: ik heb mijn eerste LaT_EX-document opgeleverd. Een simpele klachtenbrief voor het vak Nederlands (ja, dat is een verplicht vak op het hbo), dat wel, maar het was een begin.

1.2 Ervaring opdoen op Honoursprogramma

In 2018 meldde ik mij aan voor een Honoursprogramma op een dochterinstelling van Windesheim: Windesheim Flevoland te Almere. Dit Honoursprogramma, New Towns heette het, was een extracurriculair traject dat ging over zelfontwikkeling, dingen leren die niet binnen je opleiding ter sprake komen en het aanpakken van zogenaamde „wicked problems”, problemen die erg lastig zijn op te lossen omdat er zoveel betrokken partijen zijn met elk hun eigen belang. Een pittig maar leerzaam traject. Tijdens dit programma moest ik een Individueel Leertraject bijhouden, een soort dagboek over al mijn denkprocessen. Ik ben helemaal losgegaan en een bestand van 1400 pagina’s en een „gewicht” van 900 MB geschreven. Uiteindelijk heb ik om de leesbaarheid te garanderen, het bestand herschreven zodat er maar 13(!) pagina’s overbleven. Toch heb ik er veel aan gehad. Ik ben aardig vaardig geworden en ik begrijp het hele T_EX-ecosysteem nu beter. Veel code die ik vind op Internet is mij nog steeds onduidelijk, maar ik heb nu wel het idee dat ik alles kan maken wat ik zou willen maken.

1.3 Tweedejaarsstage: T_EX automatiseren

Aan het eind van schooljaar 2018-2019 moest ik een stage doen over webontwikkeling. Ik deed hem samen met een vriend van de opleiding voor Trespa. Ik denk dat dit bedrijf bij (bijna) iedereen wel een lampje doet branden. Dat is ook niet zo gek: het is een enorme speler in sierbeplating voor gevels van huizen. Zij wilden een webapplicatie waarmee je voor één van hun producten (namelijk Trespa Pura NFC) kon berekenen hoeveel platen je nodig had op een huis. Aan het einde van het programma kon je een bestelbon downloaden en – u raadt het al – dit heb ik gemaakt met T_EX.

Door X_YL_AT_EX aan de populaire programmeertaal PHP te koppelen, kon ik het genereren van bestelbonnen automatiseren. Ik had niet genoeg tijd om het systeem schaalbaar en efficiënt te maken, maar toch was ik erg trots! Het smaakte naar meer en ik zie mezelf later graag werken bij een baan waar T_EX niet (alleen) als

codeertaal wordt gebruikt om documentatie met de hand mee te schrijven, maar het (ook) als programmeertaal geïntegreerd zit in automatische systemen.

Ook bij de documentatie van dit project kon ik uitvoerig gebruik maken van \TeX . Zo moest ik mijn voortgangsrapportages bundelen en voorzien van een voorblad. Met \TeX kan dit zo elegant dat dit haast magisch wordt, je gebruikt een simpele $\text{PGF-}\backslash\text{FOR}$ -loop die één voor één alle pdfjes inlaadt en je bent klaar.

1.4 De toekomst

Ik moet toegeven dat ik nog steeds minder weet over \TeX dan ik eigenlijk zou willen. Veel dingen vind ik nog steeds erg lastig, zoals begrijpen wat de ingewikkelde macro's doen die je wel eens op \TeX Answers tegenkomt. Ook ben ik nog steeds benieuwd naar wat voor mogelijkheden $\text{Lua}\TeX$ heeft omtrent het schrijven van automatische scripts in mijn documenten. Dit alles wil ik in de toekomst nog een keer oppakken. Mijn uiteindelijke einddoel is dat ik zelf \TeX -packages kan schrijven en steeds meer van de dingen die ik nu met binaire bestanden doe (zoals afbeeldingen) oplos met code. De reden hiervoor staat beschreven in subsectie 2.1. Dit gaat nu al erg goed: ik upload mijn UML-diagrammen niet meer als afbeelding, maar als code van speciale diagramcodeertalen.

2 Mijn tools naar keuze

2.1 Toolchains

Frans vroeg mij ook om een beschrijving van mijn gebruikte tools. Welnu, het interessante deel van mijn *setup* is niet wat ik in \TeX gebruik, maar meer wat er aan de buitenkant zit.

Aan de basis gebruik ik Windows. Hierop draai ik Chocolatey, een zogeheten package manager voor Windows. Hiermee kan je in een commandoprompt programma's installen en updaten. Het voordeel is dat je met één commando alle programma's op je computer kan updaten. Elke ochtend zit ik dus op de meest recente versies van alle software die ik op mijn computer heb.

Ik gebruik $\text{Mik}\TeX$ als \TeX -distributie en \TeX studio om mijn code in te schrijven. Ik ben niet helemaal fan van $\text{Mik}\TeX$, het is naar mijn mening nogal traag als je meer packages op je systeem hebt staan. Ik moet ooit nog eens \TeX Live proberen, omdat dat op Linux-machines altijd best wel snel draaide, ook met veel packages erop. Ik moet alleen even kijken of het niet vreselijk misgaat als ik twee \TeX -distributies tegelijk installeer voordat ik het ga proberen.

Een veelgehoord argument tegenwoordig om \TeX te blijven gebruiken, ook al is Word door de jaren heen sterk verbeterd, is dat \TeX onder versiebeheer kan

worden geplaatst. Deze krachtige tools zijn voor software-ontwikkelaars gemeengoed, voor anderen is er misschien iets meer uitleg nodig. Ik heb geprobeerd het op een korte en bondige manier uit te leggen, maar dat is helaas niet gelukt. Maar probeer het maar te zien als een soort Dropbox of Google Drive, maar dan op zo'n manier gedaan dat je heel nauwkeurig kan sturen hoe het project wordt gesynchroniseerd. Deze geavanceerde systemen werken alleen echt goed met platte tekst. Daarom gebruik ik ook PlantUML in mijn code. Dit is het hierboven in subsectie 1.4 al genoemde systeem om door middel van code software-diagrammen te maken. Doordat ik dit in code doe en niet met een grafische editor, kan ik dit ook onder versiebeheer zetten. In mijn versiebeheer maak ik ook gebruik van een CI/CD-straat. Dit is een systeem dat ervoor zorgt dat de bestanden op de versiebeheerserver automatisch worden gecompileerd.

2.2 Classfiles en fonts

Ik gebruik eigenlijk alleen maar standaard classfiles en lettertypes. Standaard is goed, is mijn mening, anders hebben ze het wel anders bedacht. Soms gebruik ik KOMA-Script voor functies als $\backslash\text{part}$. Als ik Unicode-karakters moet schrijven in $\text{X}\TeX$ en ik het bestand ook op een Linux-machine moet kunnen gebruiken, gebruik ik FreeSans. En als ik echt een mooie romantische brief moet schrijven, gebruik ik het Uwebo-dingbatfont. Ziet er fantastisch uit!

3 Werkwijze

Veel van mijn vrienden op de universiteit krijgen les in \TeX . Ik niet, ik heb mijzelf alles aangeleerd. Wikibooks en Overleaf bieden hiervoor heel goede documentatie en de leercurve is ook zeker niet te steil. Als ik iets niet weet, dan kopieer ik het vaak van \TeX - $\text{La}\TeX$ Stack Exchange. Dit zorgt ervoor dat ik vrij veel kan, maar jammer genoeg zonder echt te weten hoe het allemaal werkt.

4 \TeX in mijn omgeving

Zoals hiervoor al gezegd, heb ik \TeX mijzelf aan moeten leren. Op het HBO op „Windowsheim” zijn Microsoft Word en Google Docs erg populair. In theorie integreert het perfect met het grote intranet en kan je naadloos samenwerken, maar in de praktijk is het erg brak. Enkele studenten en docenten zijn erg enthousiast over \TeX . Soms leer ik mijn docenten ook nog eens wat! Andere mensen vinden het maar een gedoe om een extra taal te leren. En een enkeling vindt het wel oké om samen te werken, maar is er niet superenthousiast over.

5 Mijn visie op T_EX

T_EX is een erg leuk systeem en het is ook leuk dat er zo'n trotse community voor beschikbaar is. Vooral de artikelen in de TUGboat over onderzoek in typografie vind ik erg interessant. En het compileren van een document heeft altijd wat moois, ook al is het jammer dat het vaak zo lang duurt. En het eerder genoemde voordeel, dat je een heel project in code kan schrijven, is een prachtig fenomeen.

Maar T_EX voelt wel primitief. Het is duidelijk niet gebouwd om er echt stevig in te programmeren. Veel codevoorbeelden op T_EX - LaT_EX Stack Exchange geven me nog steeds hoofdpijn. Er zijn alternatieven, zoals Markdown en `wkhtmltopdf`, die beide de uitgebreide programmeertaal Javascript ondersteunen. Toegegeven, Javascript heeft ook zijn eigenaardigheden, maar simpele dingen zoals arrays zitten er wel gewoon in. Maar als ik die gebruik, is er geen uitdaging meer om dingen op te lossen in echte (L^A)T_EX. En juist die uitdaging vind ik leuk. Bovendien krijg ik dan ook niet de prachtige output van T_EX. Misschien is Lua(L^A)T_EX een oplossing?

Verklarende Woordenlijst

applicatie een stuk software. Hoeft niet per se een app te zijn voor de smartphone of tablet, maar kan ook een computerprogramma of website zijn.

array lijst variabelen.

codeertaal een taal die lijkt op een programmeertaal, maar niet aan alle voorwaarden voldoet om een programmeertaal te zijn. Een voorbeeld is HTML.

Linux familie van veelal gratis alternatieven op het Windows-besturingssysteem van Microsoft.

UML Unified Modeling Language, een diagramstandaard voor het beschrijven van software.

Dennis Holierhoek
24 februari 2020