

Stroomdiagrammen maken met flow

Abstract

Flow is een handig programma om stroomdiagrammen te maken; via `\write18` kan het vanuit een LaTeX-document worden aangeroepen zodat eventuele wijzigingen vanzelf tevoorschijn komen.

Keywords

Latex, flow, write18, stroomdiagram, softwaredocumentatie.

Hoe maak je stroomdiagrammen met flow?

Als onderdeel van het vorige artikel wilde ik graag een schemaatje opnemen. Ik stuitte bij het zoeken naar een eenvoudig programma voor het maken van stroomschema's op `flow`¹, een 'taaltje' ontwikkeld door *Terry Brown* dat is voorzien van alle basiselementen van een stroomdiagram, maar vooral van een aantrekkelijk eenvoudige syntax (zie pagina 18 van *The LaTeX Graphics Companion* Goossens, Rahtz, Mittelbach, 1997).

De beschikbare elementen zijn: `Box`, `Oval`, `Choice`, `Tilt` en `Text`. Het laatste element is een box zonder rand.

Het is voor de opzet van een diagram niet nodig coördinaten op te geven voor de grootte van de elementen, maar voor de afmetingen en verplaatsingen die wél afwijken van de standaard moet je getallen invoeren.

De vorm van de elementen (grootte en hoogte/breedte-verhouding) kun je ook met eenvoudige getallen bepalen. Het is belangrijk vooraf de eenheid voor die getallen vast te leggen, bijvoorbeeld met:

```
\setlength{\unitlength}{2em}.
```

In de elementen kun je – al dan niet gecentreerd – tekst opnemen, en alle attributen die je in LaTeX kunt gebruiken zijn ook in flow bruikbaar. Vanzelfsprekend is het nodig er op te letten dat de tekst die binnen de randen van een box moet worden weergegeven niet te groot is, met andere woorden: de grootte van de box en de grootte van de tekst moeten op elkaar afgestemd zijn.

Er zijn drie types voor de verbindinglijnen tussen de elementen, namelijk `line`, `arrow` en `none`, die kun je instellen met commando `Settrack`. De richting in het schema geef je aan met respectievelijk `Right`, `Left`, `Down` en `Up`. Door te manipuleren met de `\unitlength` is een afbeelding eventueel te schalen:

```
\newlength{\um}\setlength{\um}{1bp}
\setlength{\unitlength}{15\um} % gebruik 19\um voor storm.tex.
```

Doorgaans moet het font in het stroomdiagram dan meegeschaald worden, bijvoorbeeld door `\scriptsize` te kiezen in plaats van `\normalsize`. Het resultaat is een picture dat zich naadloos laat opnemen in het LaTeX-document.

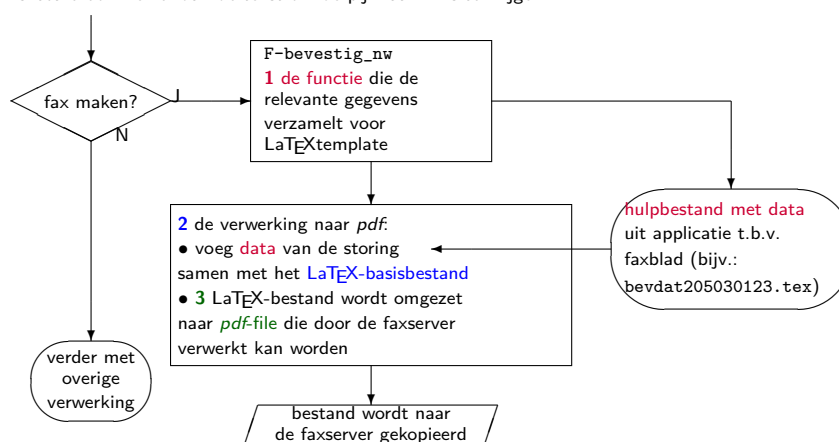
Vooraf ter illustratie een stukje flow:

```

TxtPos [l] [l] ~~
Text 6 1
  onzichtbare box kleiner dan de tekst om de pijl naar links te krijgen
Down .15
SetTrack Arrow
TxtPos [c] [c]
Choice . . J N 4 2
  fax maken?
Tag
Down 4
Oval 3 2

```

onzichtbare box kleiner dan de tekst om de pijl naar links te krijgen



Met een Tag en ToTag kun je herhalingen construeren, of ‘teruggaan’ naar een aftakking. Een en ander is in flow.pdf voldoende gedocumenteerd.

Om dit goed werkend te krijgen moesten er wel een paar hindernissen genomen worden...

Hoe krijg je flow aan de praat?

Ik beschrijf de stappen die ik achtereenvolgens moest zetten. Let wel: ik werk doorgaans op een pc met Windows-XP dus de voorbeelden zijn daaraan ontleend:

- bereid aan de hand van een voorbeeld uit de documentatie een proefdocumentje bijvoorbeeld proef1.flo
- uitproberen, vanaf de commandline: flow proef1.flo proef1.tex
- het resultaat is een stukje picture-code, dat in principe meteen ingevoegd kan worden in je document
- als blijkt dat in de eerste regel van het op deze manier gemaakte plaatje het volgende staat:


```
\begin{picture}(2989276266496.000000,2.000000)(0.000000,-2.000000)
```

 dan levert dat bij verwerken van het picture een foutmelding op; oplossing is:
- flow.exe upgraden naar 0.99e; de laatste versie die voor gebruik in MS-dos/-

Windows kant en klaar beschikbaar was is versie 0.99b

- er zit in dit geval niets anders op dan met een geschikte compiler `flow.c` zelf te compileren (ik gebruikte MYSYS en Mingw); hoe dit gaat valt buiten het bestek van dit artikel.

Voor een ander doel had ik al vaak de behoefte gehad om tegelijk met een LaTeX-run een extern commando te laten uitvoeren – als bijvoorbeeld op een inhoudsopgave een externe bewerking gedaan moet worden – en daarvoor zocht ik naar een bruikbare en ‘nette’ oplossing.

Voor dit artikel moet na iedere *wijziging* van `schema2.flo` het flow-diagrammetje opnieuw omgezet worden naar het picture.

Dit is mogelijk met een `\write18{<externe opdracht>}`; de externe opdracht kan ieder commando zijn dat normaal op de command line van Windows of Linux ingetypt wordt.

Het is wel nodig eerst `write18` te activeren² door in het bestand `texmf.cnf` de parameter `shell_escape` op `t` te zetten:

- MS-Windows-gebruikers moeten eerst het `texmf.cnf`-bestand zien te vinden – waarbij een probleem is dat in windows de extensie `.cnf` hardnekkig als een snelkoppeling wordt weergegeven
- bij het gebruik van de software van de TeX-live-cd zijn er twee exemplaren van `texmf.cnf`; ik heb ze allebei gewijzigd (één van beide heeft ‘voorrang’) en dat leverde het gewenste resultaat op: de log-file meldt dat `write18` enabled is
- MS-Windows-gebruikers die Miktex van de TeX-live-cd hebben geïnstalleerd (TeXnicCenter) moeten de Miktex-equivalent van `texmf.cnf` zien te vinden: `miktex.ini`.

Nu gaat het erom het juiste commando te maken dat via `write18` uitgevoerd kan worden: `flow.exe schema2.flo schema2pr.tex`.

De aanroep van flow gaat dan als volgt:

```
\write18{flow.exe schema2.flo schema2pr.tex}.
```

Met dezelfde LaTeX-run wordt het hiermee gemaakte ‘plaatje’ in de tekst opgenomen: `\input schema2pr.tex`. Op deze manier kan het stroomschemaatje gelijktijdig met de rest van het document gewijzigd worden en worden de laatst opgeslagen wijzigingen meteen in het eindresultaat verwerkt.

Op www.ctan.org/tex-archive/support/flow zijn source en documentatie van flow te vinden.

Voetnoten

1. een standaard-C programma voor het maken van stroomdiagrammen in de LaTeX picture-omgeving.
2. de shell-command optie staat standaard uit uit veiligheidsoverwegingen (in `miktex.ini`: `write18=disable`).

Ernst van der Storm
evdstorm@msso.com