tool view

Albert Mietus'

toolview

Elke ingenieur gebruikt tools en iedereen heeft zijn eigen voorkeur; vaak gebaseerd op wat er toevallig als eerste beschikbaar was. Zo ook de software ingenieur. Ook hij gebruikt tools, software-tools wel te verstaan. Daarvan zijn er oneindig veel, waar-door kiezen erg moeilijk is. Veel handige tools worden niet gebruikt, eenvoudigweg omdat niet bekend was dat er een tool is, specifiek voor dat probleem.

Het leren gebruiken van een tool kost tijd. Als je de naam en het doel van het juiste tool kent, loont het gebruik vaak al de moeite. Het inleren en inzetten kost vaak minder tijd dan aanmodderen zonder goed tool.

'Dot' is een taal, maar ook een tool om graven te tekenen. Het is onderdeel van het 'graphviz' pakket. Ken jij ook een handig pakket, schrijf dan ook eens tool(re)view.

Mail me, voor meer info
ALbert dot Mietus at PTS dot nl

door & more

Graphviz

De tools van Graphviz zijn primair ontwikkeld voor Unix systemen, door één van de godfathers van Unix: AT&T. De sourcecode is beschikbaar. Er zijn ook binaire versies beschikbaar, waaronder voor windows.
Omdat het een research project is, moedigt AT&T het gebruik van dit tool aan; zelfs voor commercieel gebruik! De open-source wereld heeft, net als AT&T zelf, vele nuttige en minder nuttige uitbreidingen bedacht. Zo zijn er web-versies van het tool. Leuk!

Duidelijk is dat het tool vooral bedoeld is voor onderzoek. Zoek je een mooi tool, zoek dan vooral verder.

Voor ontwikkelaars is de site www.graphviz.org interessant; echter de officiële site is www.research.att.com/sw/tools/graphviz.

DOT-Graphviz

Iedereen weet hoe belangrijk bomen zijn. Voor de informaticus zijn bomen echter een bijzondere vorm van graven. Een boom heeft takken en bladeren. En knopen, althans voor ICTers. Het lijkt wel natuurkunde maar toch is graventheorie onderdeel van de wiskunde. Het zijn dan ook wiskundigen die 'graphviz' geschreven hebben. Waarbij het tool 'dot' heel snel inzicht kan geven in een complexe boom.

Saai of mooi?

Het is een saai tool, waarmee je mooie plaatjes kunt maken. Plaatjes van relaties; zoals tussen (include)files. Of tussen functies, classes of desnoods netwerk-nodes. Want zelfs een klein programma heeft al snel een grotere 'call-tree' dan we kunnen bevatten. En daarnaast een 'include-tree', een file-hiërarchie en talloze ander boomachtige structuren. De relaties zijn stuk voor stuk in de (programma)tekst beschreven. Door die grafisch weer te geven, krijgen we vaak meer inzicht. Zeker als de resulterende plaatjes groot zijn. Handmatig tekenen is veel werk. En het bijhouden daarvan nog meer. Dankzij dot kunnen we dat automatiseren. Als we een scriptje schrijven dat die relaties stuk voor stuk bijeen 'grept' kunnen we daarmee een heel redelijk plaatje genereren.

Automatische plaatsing

Graphviz bestaat uit meerdere tooltjes, één voor elke soort graaf of bewerkingen daarop. En allemaal gebruiken ze de taal 'dot', met een heel eenvoudige syntax om elementen en hun relaties vast te leggen. Inclusief optionele attributen als kleur en tekenstijl. Ook is er een tool genaamd 'dot'. Dat leest een dot-file met (slechts) knopen en takken en vertaalt dat in een plaatje. De knopen en takken worden automatisch geplaatst. Het plaatje kan in diverse formaten weggeschreven worden; zoals png en gif. Ook kan het plaatje op het scherm getekend worden, of kan een nieuwe dot-file geschreven worden, waarbij plaatsing en tekenattributen toegevoegd zijn.
Met de andere tools kunnen cyclische graven getekend worden. Of kan een graaf vereenvoudigd worden. Met 'dotty' kan je zelfs interactief een plaatje veranderen. Leuk! Maar erg beperkt als je moderne teken-programma's gewend bent.

Visualiseer

Met bijvoorbeeld een awk-scriptje kun je een dot-file eenvoudig genereren. Bijvoorbeeld door te grepen op '#include'. Waardoor de include-boom getekend kan worden. Door de output van een tool als cscope te converteren, kun je de complete call-tree van een applicatie visualiseren. Vooral bij software die je nog niet kent, kan zo'n plaat een hele goede indruk geven van de complexiteit. Veel sneller dan door alleen de kale code te bestuderen. Ook andere relaties kunnen gevisualiseerd worden. Zo is de relatie tussen de verschillende Unix-varianten een standaard voorbeeld.

Complexiteit

Stel je wilt een complexe structuur zien. Zodra je weet hoe je losse relaties uit 'de code' kunt trekken, kan er je er een plaatje mee beschrijven en dus visualiseren. Dat kan zelfs een datastructuur zijn. Je kunt het zo complex maken als je wilt. Eindelijk kun je iets complex mooi weergeven. Is dat saai of is dat mooi? Of is het mooi dat je complex haast saai kunt maken? Probeer het maar. Spannend!