

De scriptrevolutie

Marianne van den Boomen

in: *I & I, Nieuw eMedia in Perspectief*, nummer 5, 2002

Hoewel ik onder elke column altijd mijn e-mailadres zet, is het aantal aantal lezers dat reageert toch altijd een beetje teleurstellend. Om eerlijk te zijn: het gebeurt zelden. En ik mail toch echt altijd keurig terug - maar ja, dat kunt u van te voren niet weten. Sommige reacties verdienen echter een hele column, zoals nu.

Ik schreef in het vorige nummer van I&I (http://www.xs4all.nl/~boom/artikel/nieuw/art_I&I_multimedia.htm) over hoe, ondanks alle multimediahype, het web nog altijd vooral een tekstueel medium is. Niet om het web af te doen, maar om het idee van wat tekst nu eigenlijk is, op te waarderen. Tekst is veel veelzijdiger en multimedialer dan wij denken. Ik stelde daarbij ook dat de echte doorbraak van het web bestond uit de radicalisering van wat tekst vermag. Oftewel, de scriptrevolutie: de opkomst van allerhande scripts en scripttalen (Javascript, PHP, ASP etcetera) die dynamische en interactieve HTML-pagina's mogelijk maken.

Een oplettende lezer vond dat de passage over scripts het verhaal ondermijnde. Hij schreef: 'In de WWW context is een script gewoon een programmaatje. Dat het er bij inspectie uitziet als tekst, is niet relevant. Want ook als je een programma geen script noemt, ziet het er meestal uit als tekst. Dat scripts worden gebruikt om tekst mee te dynamiseren is ook niet relevant, want ze zouden ook kunnen worden gebruikt om beelden te manipuleren. Sterker nog, de term script, of scenario in goed Nederlands, is afkomstig uit de wereld van de bewegende beelden en actoren, toneel en film.'

Deels heeft hij gelijk. Inderdaad, op een paar punten komen programma's en scripts overeen. Ze zien er beide uit als tekst, ze zijn beide geschreven in een specifieke taal en ze bestaan beide uit een al dan niet complexe serie als-dan-instructies en *loops*.

Maar het gaat er niet om hoe het eruitziet, het gaat erom wat het is en hoe het werkt - en daarin verschillen programma's en scripts fundamenteel. Een programma is een self-sufficient totaalpakket aan code dat, mits juist gecompileerd, geïnstalleerd, geconfigureerd en opgestart, de computer een serie handelingen of bewerkingen kan laten uitvoeren, al dan niet op commando van de gebruiker. ('t Is een ingewikkelde mondvul, zo'n definitie, maar dat spoort aardig met de praktijk...) Een goed geïnstalleerd programma verbouwt in feite de computer tot een specifieke machine met specifieke functies en mogelijkheden. Het zet als het ware de radertje en tandwielletjes van de hardware, met name het geheugen, paraat in de juiste stand.

Programma's zijn dan ook meestal platformafhankelijk; afhankelijk van je besturingssysteem (Windows, Mac OS, Linux) is een andere versie nodig. Afgezien daarvan merk je tegenwoordig nauwelijks meer iets van dat omzetten van de hardware-radertjes - behalve als specifieke combinaties van opgestarte of zelfs geïnstalleerde programma's elkaar in de weg zitten en dus crashes veroorzaken; ook computerradertjes kunnen op één moment maar één stand tegelijk hebben... Wie ooit begin jaren negentig een modem heeft geïnstalleerd, herinnert zich nog wel hoe je toen daadwerkelijk radertjes (jumpers of switches) met de hand c.q. een pincet moest omzetten - dat deed de software nog niet geheel zelf.

Een script is een relatief autonoom onderdeel van een programma: een semi-zelfstandige module van direct leesbare en uitvoerbare broncode. Het gaat hier om apart beheerbare subroutines binnen een specifiek programma, zoals bijvoorbeeld zelfgemaakte macro's in tekstverwerkers. In feite zijn ze parasitair: ze hebben een programma nodig om gelezen en uitgevoerd te worden. Dat geldt ook voor scripts op websites: de scriptcode, die is ingebed in de HTML-code, heeft de browser van de gebruiker nodig om te kunnen werken, en die browser moet de betreffende scripttaal kunnen lezen. Een gemiddelde browser kan dat tegenwoordig (met uitzondering van sommige braillebrowsers), en dat betekent dat gebruikers geen omkijken hebben naar scripts op websites. Ze hoeven niks te installeren of op te starten, hun computer hoeft niet anders geprogrammeerd te worden, het maakt niet uit wat voor platform ze hebben - hun browser leest gewoon het script en voert het uit.

Dat is het grote verschil tussen een programma en een script: een script werkt als tekst, het komt tot leven door lezen zonder meer, zij het dat het lezen door de computer gebeurt. Bij een programma moet heel wat meer gebeuren dan slechts lezen om het te laten werken: installeren (zorgen dat alle onderdelen van het programma op de juiste plek op de computer en in het geheugen zitten), configureren (specifieke instellingen bepalen) en opstarten (in het werkgeheugen laden). Pas dan kan het programma zijn eigen code lezen en uitvoeren. Dat dit zo werkt, heeft alles te maken met het feit dat een script, anders dan een programma, bestaat uit direct leesbare *broncode*. Broncode is de code waarin de programmeur zijn of haar programma schrijft. Die broncode moet in het geval van programma's vervolgens worden omgezet ('gecompileerd') met een omzetprogramma ('compiler') tot object code, dat wil zeggen: door de machine leesbare code. Alleen van scripts is de broncode direct uitvoerbaar door een machine; compileren is daar niet nodig. Dat typeert een tweede, zo mogelijk nog crucialer verschil tussen programma's en scripts: aangezien programma's gesplitst zijn in broncode en gecompileerde objectcode, zijn ze gemakkelijk *proprietary* te houden - simpel door aan gebruikers alleen de object code te geven. Meer hebben ze toch niet nodig om te installeren. De broncode is dan fabrieksgeheim, en alleen de eigen fabrieksprogrammeurs kunnen erbij om verdere toepassingen te maken. Microsoft bijvoorbeeld gaf slechts wat broncode van Windows vrij na allerlei anti-monopolierechtzaken. Zogeheten open source programmatuur, zoals Linux, geeft principieel naast de objectcode ook de broncode erbij, maar dat is bepaald niet de default-situatie van programmatuur. Je zou kunnen zeggen dat programma's, door hun specifieke aard, ontologisch *closed source* zijn: een gesloten systeem van ondoordringbare, onveranderbare object-code, zonder bijbehorende broncode. Scripts daarentegen zijn naar hun aard ontologisch *open source*: een open systeem omdat de broncode per definitie meekomt.

Het moge hopelijk duidelijk zijn dat ontologische open of closed source iets anders is dan juridische open en closed source. Programma's kunnen juridisch open source zijn, scripts kunnen auteursrechtelijk beschermd zijn en vice versa. Waar het om gaat, is dat - los van hoe de zaken juridisch geregeld zijn - scripts naar hun aard opener zijn dan programmatuur. En precies dat is kracht van de scriptrevolutie op het web. Immers, hoe opener, transparanter en bewerkbaarder de code is, hoe meer publieke versies en varianten er worden gemaakt door creatieve nerds en luie cut-and-paste-scriptkiddies. Ja helaas, daar komt ook de virusexplosie van de laatste jaren vandaan; virusproductie is geen ambachtelijk programmeurswerk meer maar een kwestie van een beetje editten in simpele scripts. Juist omdat scripts niet geïnstalleerd hoeven worden en parasiteren op reeds draaiende programma's, zijn ze als attachment zo verraderlijk. Lezen = uitvoeren...

De hoeveelheid zinvolle al dan niet gratis te downloaden scripts is echter ook gigantisch (zie bijvoorbeeld www.hotscripts.com). Mooie uitvinding, teksts-scripts.

Dat de term script ook circuleert in de film- en theaterwereld van bewegende beelden en actoren, maakt het script overigens niet minder tekstueel. Ook een film- of toneelscript is de in tekst gegoten versie van de handeling, een relatief autonome subroutine van de totale uitvoering in het kader van een programma. Er komen alleen geen virussen van...