

XML en crossmedia

Roelof Janssen

Screens & Pages

R.J.Janssen@SandP.nl

Inleiding

De term crossmedia wordt op verschillende manieren geïnterpreteerd; er zijn zowel benaderingen vanuit de techniek als vanuit de communicatieve wetenschap. Het woord crossmedia wordt vaak in relatie gebracht met termen als ‘mediumneutraal publiceren’ en ‘single source publishing’. In de jaren 90 van de vorige eeuw werd de term ‘Create Once, Publish Many’ voor het eerst gebruikt door uitgevers zoals Paul Zazzera, the CIO of Time, Inc. die als doel had om zijn informatie vanuit één bronbestand te publiceren zowel op papier als op een website.

Indira Reynaert [1] definieert crossmedia als volgt: Er is sprake van crossmedia indien er een kruisbestuiving bestaat van verschillende media zoals theater, film, televisie, radio, print, internet, games, mobiele devices en live-events, waarbij de verschillende media mediumspecifieke betekenissen communiceren welke deel uitmaken van een overkoepelende boodschap.

Monique de Haas [2] ziet crossmedia als het opbouwen van een verhaal dat via verschillende media verloopt: “Bij crossmedia communicatie nodigt het verhaal je uit om een cross-over van het eerste medium naar het volgende te maken. Zo verandert de communicatie van eenrichtings- naar tweerichtingsverkeer.”

Nieuw is dat inmiddels ook real-world projecten worden bedacht en uitgevoerd waarin deze principes worden toegepast, met name in de reclamewereld. Print, web en mobiele telefoon worden in een vooraf uitgedachte volgorde gebruikt om de gebruiker tot een (koop)actie te verleiden.

Om te voorkomen dat content voor ieder medium apart ontwikkeld moet worden is mediumneutrale creatie, opslag en verwerking een belangrijke pijler onder een crossmedia project. Een voor de handliggende keuze is dan het gebruik van XML, of als content‘store’ of als tussenformaat om de verschillende media-uitingen te genereren. Een belangrijk voordeel bij het gebruik van XML is dat de selecties gemaakt kunnen worden waarmee het mogelijk wordt om

informatie te filteren, te sorteren etc. Verder zijn platformafhankelijkheid en scheiding van vorm en inhoud in deze context belangrijk omdat er - zeker in de wereld van de mobile readers – veel hardware- en operating systeem varianten bestaan.

XML in de creatieve omgeving

Reclamebureaus, game- en 3D ontwikkelaars, webbouwers en AV bedrijven spelen een belangrijke rol als het gaat om crossmedia projecten. Daarbij richt men zich hoofdzakelijk op de interface en de content, die met name ‘cool’ en in ieder geval gebruikersvriendelijk moet zijn. XML en aanverwante standaarden zoals X-Forms waren daar voorheen niet populair; het meest gebruikte tool is Flash van Macromedia, dat inmiddels door Adobe is ingelijfd.

Adobe is wel een speler die dit kan beïnvloeden; het levert nu zowel software voor grafische toepassingen (Adobe InDesign, Adobe Illustrator, Adobe PhotoShop) als voor websitebouw (Dreamweaver en GoLive!) , animatie (Flash), digitale formulieren (Acrobat, Flex) en audio/video.

In toenemende mate wordt in deze producten XML ondersteuning ingebouwd, zowel voor metadata (XMP) als voor import en export van XML. Met name voor crossmedia wordt veel verwacht van Flash en Flex in combinatie met XML. Voor Adobe is de opkomst van mobiele telefoons die Flash Lite ondersteunen waarschijnlijk een belangrijke factor geweest bij de overname; dit is een ‘uitgeklede’ versie van Flash die het mogelijk maakt om binnen de beperkingen van het kleine telefoonscherm toch een mooie en snelle interface te kunnen bieden. In 2004 konden 17.2 miljoen telefoons dit aan, eind 2005 zullen dat er 40 miljoen zijn. Het gebruik van Flash Lite is met name populair in Japan; NTT DocoMo levert daar aan 2 miljoen gebruikers van i-channel diensten met nieuws en informatie via de mobiele telefoon. Daarbij gebruiken zij de Adobe FlashCast techniek op hun servers. Verder biedt Flash een goede ontwikkelomgeving voor mobile games omdat de vectorgebaseerde content goed aan te passen is aan de pixelafmetingen van het display.

Daarnaast gaat Sony ook Flash ondersteunen in hun Playstation Portable gameconsole.

Flash Communication Server is een belangrijk product voor het streamen van video via het web en wordt veel gebruikt bij omroepen en beeld- en muziek leveranciers zoals het Nederlandse FabChannel.

Flash en XML

De voorkeur van creatieven voor het gebruik van Flash op websites en mobiele apparatuur (er schijnt zelfs een ijskast met een Flash display te zijn) is verklaarbaar vanwege de volgende eigenschappen:

- De ontwerper heeft (bijna) volledige controle over de positie, het uiterlijk en de bewegingen van de door hem bedachte objecten.

- Na een zekere laadtijd (steeds minder een probleem vanwege de goede penetratie van breedband in Nederland) reageert een Flash 'movie' snel op handelingen van de gebruiker en zijn spectaculaire effecten mogelijk zoals snel in/uit zoomen, afspelen van videoclips, genereren van geluiden etc. In die zin wordt Flash ook wel gezien als een RIA (Rich Internet Application) toepassing 'avant la lettre'. De afhandeling van objecten en tekst als een vectordefinitie plus de goede compressie van videomateriaal zorgen voor relatief compacte bestanden met toch veel impact.

- Bij Flash is er een verschil tussen het 'source' bestand (.fla) en het afspeelbare bestand (de 'executable'), waarmee de professionele maker zich beschermt tegen het knippen en plakken van de door hem ontworpen objecten, scripts en effecten. Diezelfde eigenschap heeft Flash juist impopulair gemaakt bij ICT-ers en webdevelopers die inhoud en vorm extern willen aansturen of delen willen hergebruiken zoals dat bij JavaScript wel kan. Grappige bijkomstigheid is dat SVG, dat beschouwd kan worden als een 'open' alternatief voor Flash, ook uit de koker van Adobe komt (maar toen hadden ze Macromedia nog niet overgenomen).

Ieder nadeel heeft zijn voordeel: juist door de geslotenheid van een Flash movie (dat vanuit ICT oogpunt het kan worden gezien als een BLOB) en de goede penetratie van de Flash Player in de meeste webbrowsers (en in sommige PDA's en mobiele telefoons) werkt het vaak wel zoals de ontwerper bedoeld heeft. Daarbij speelt er wel een versieprobleem: de nieuwste versie van de FlashPlayer (versie 9) is nog niet wijdverbreid en zal door velen eerst gedownload moeten worden om de nieuwe functies te kunnen gebruiken.

Programmeurs kunnen binnen Flash gebruik maken van een JavaScript-achtige scripttaal, ActionScript. Op die manier kan bijvoorbeeld tekst uit een database worden ingelezen en dynamisch worden omgezet in een 'flying logo'. Dit is ook de sleutel naar het gebruik van XML in Flash. Sinds enkele jaren kent ActionScript een XML class die het mogelijk maakt een XML bestand te lezen en door de nodes daarin te wandelen. Op die manier wordt het gesloten karakter

van Flash enigszins opengebrouwen en kunnen dus teksten en allerlei andere attributen via een extern bestand variabel gemaakt worden [3].

Overigens is de ondersteuning vrij basaal; controle op basis van een DTD of uitvoering van XSLT binnen Flex is niet mogelijk, dat zal in de fase daarvoor geregeld moeten worden.

FlexBuilder

Voor de meer zakelijke toepassingen waarbij data-entry en digitale formulieren een rol spelen heeft Macromedia indertijd FlexBuilder ontwikkeld die uit twee deelproducten bestaat, de FlexBuilder IDE (Integrated Development Environment) en het Flex Framework. Verder is de Flash Player 8.5 nodig om de resultaten af te kunnen spelen in een browser zoals Internet Explorer, Firefox of Netscape/Mozilla.

FlexBuilder maakt zoveel mogelijk gebruik van standaarden: de IDE is ontwikkeld in Java en de broncode van applicaties bestaat uit XML (met een eigen namespace); Adobe noemt dit MXML.

Bij het gebruik van FlexBuilder valt direct een vergelijking op met de Microsoft .NET omgeving; ook daar kan geschakeld worden tussen de visuele representatie en de broncode, die nauw samenwerken. Na het plaatsen van interface componenten zoals tekstvelden, knoppen, links en drop down menu's kan de samenwerking daartussen worden geregeld via eigenschappen of door in de broncode regels XML toe te voegen (zie afbeeldingen). Het resultaat kan worden gecompileerd (er wordt dan een HTML bestand plus een SWF file aangemaakt) en in een browser worden getest. Het eindresultaat blijft dus een Flash object in een HTML bestand maar het grote voordeel is dat in vrij korte tijd ingewikkelde functionaliteit gemaakt kan worden. Binnen een paar minuten kan men leuke toepassingen maken, zoals het uitlezen van de RSS feed van een weblog, plaatsen daarvan in een lijst via een DataGrid en ophalen van het totale bericht met een muisklik.

Op deze manier wil Adobe het werk dat traditioneel door een programmeur gedaan moet worden ook binnen het bereik van meer creatieve webbouwers brengen.

OpenLazlo

De werkwijze van Flex wordt ook gevolgd bij het Open Source project OpenLaszlo; ook dit genereert een Flash bestand op basis van scripts die in XML zijn gespecificeerd. De ontwikkeling

wordt gesponsord door LaszloSystems. IBM Alphaworks heeft als bijdrage ook een IDE hiervoor ontwikkeld, die net als Flex gebaseerd is op Eclipse [4].

Het principe is dat de interactiviteit wordt bepaald door XML scripts die op de webserver worden geplaatst. Browsers roepen deze in runtime aan; de aanroep wordt door de server omgezet naar een Flash file en naar de browser teruggestuurd. De browser dient dus wel de Flash Player aan boord te hebben. Een belangrijk uitgangspunt voor datadriven applicaties is dat alleen XML als dataformaat wordt ondersteund. Bij gebruik van een database zal men dus zelf voor de omzetting naar XML moeten zorgen, bijvoorbeeld via Java Server Pages.

XML en gedrukte media

Als een uitvloeisel van database publishing is door een aantal leveranciers al ruime ervaring opgebouwd bij de omzetting van XML naar een folioprodukt.

Daarbij wordt in de meeste gevallen uitgegaan van content in de vorm van een XML document, waarin tekst en referenties naar afbeeldingen volgens een logische structuur aanwezig zijn. Mocht de informatie niet direct al op die manier beschikbaar zijn dan is het voor een programmeur meestal weinig moeite om vanuit een database, tekstverwerker of website een dergelijk XML document te genereren. In de markt zijn momenteel drie benaderingen te onderscheiden:

1. XML naar PDF via XSL-FO (Formatting Objects)

Vanuit een XML bestand kan via XSLT een FO bestand worden aangemaakt. XSL-FO is een W3C standaard die de opmaak van een document beschrijft in de vorm van tekstflows en eigenschappen van foliobegrippen zoals pagina's, kaders, kop- en voetregels, inhoudsopgave etc.

Dit FO bestand kan worden aangeboden aan een zogenaamde 'rendering engine' die daarvan een opgemaakt bestand maakt dat kan worden opgeslagen in de vorm van PDF, RTF of ander formaat. Daarbij is de informatie dan ingedeeld in pagina's, zijn afbeeldingen geplaatst, verschillende lettertypen toegekend en zijn bijvoorbeeld kop- en voetregels en paginacijfers toegevoegd.

Er zijn zowel open source renderers als commerciële FO formatters beschikbaar. De ervaring leert dat de open source producten (bijvoorbeeld Apache FOP) meer beperkingen hebben dan de commerciële producten, waaraan ook vaak een stevig prijskaartje hangt. Zo is het met FOP niet mogelijk om woorden af te breken, wat een probleem kan worden bij langere documenten

waarvan de regels uitgevuld (in blokvorm) gezet moeten worden. Ook zijn er beperkingen aan de bestandsformaten van afbeeldingen die worden ondersteund (geen EPS, geen CMYK), en ontbreken functies voor verwerking richting drukwerk zoals het werken met steunkleuren, afdrukken met snijtekens en controlestrips etc. Installeren van fonts is niet zo eenvoudig als een DTP operator gewend is. Ook kunnen toevoegingen aan een PDF resultaat zoals metadata, annotaties, navigatie etc. niet direct binnen FOP worden aangebracht.

Een fundamenteel probleem van FO is dat het in eerste instantie ontwikkeld is voor langere documenten met een tekstflow-achtige opmaak. Lay-outs met veel kaders (denk aan een retail- of makelaarskrant, educatief materiaal met veel afbeeldingen en uitleg, een vrijer vormgegeven brochure of catalogus) zijn hiermee niet of heel moeizaam te maken.

Maar voor goed gestructureerde, rapportachtige producten is XSL-FO een snelle, goede en betaalbare methode (mits de benodigde kennis beschikbaar is). FO wordt vaak toegepast bij web-to-print en e-commerce toepassingen, geautomatiseerde productiestraten bij uitgevers etc.

2. XML naar PDF met behulp van eigen software

Een aantal van de beperkingen van FO kan worden vermeden door zelf software te ontwikkelen voor de omzetting van XML naar PDF. Beide formaten zijn open en gedocumenteerd dus dit moet te programmeren zijn. Omdat PDF een complexe bestandsstructuur heeft is het dan wel verstandig om gebruik te maken van een software bibliotheek, zoals bijvoorbeeld PDFlib.

Met de nodige inspanning heeft men op die manier meer controle over de omzetting van XML (of XML-FO) naar PDF dan bijvoorbeeld met FOP. Er kunnen nu ook kaderachtige documenten gemaakt worden, waarvan delen bijvoorbeeld te personaliseren zijn door een koppeling te leggen met een tweede bestand waarin alle variabelen staan (bijvoorbeeld een spreadsheet, XML bestand of database).

De programmeur bepaalt nu zelf hoe de pagina's en de PDF's eruit zien dus snijtekens, controlestrips, orderinformatie, metadata en annotaties zijn nu ook mogelijk. De beperkingen hangen alleen af van de gebruikte software bibliotheek, de inventiviteit van de ontwikkelaar en diens kennis van grafische aangelegenheden. Punt van aandacht is dat bijvoorbeeld PDFlib tot op heden geen EPS afbeeldingen ondersteunt, wat een probleem kan zijn bij het gebruik van logo's en naar CMYK omgezette afbeeldingen.

Wijzigingen in de opmaak (andere kleuren, andere fonts etc.) moeten door de programmeur worden uitgevoerd of er moet gewerkt kunnen worden met een willekeurig PDF template. Een

tweede probleem zijn woordafbrekingen; PDFlib ondersteunt deze alleen als de woorden voorzien zijn van bijzondere tekens op de positie waarop deze afgebroken zouden kunnen worden. Er zal dus een externe afbreekroutine gebouwd of ingekocht moeten worden.

Deze methode is zinvol op het moment de XSL-FO benadering tekort schiet; er kunnen nu bijvoorbeeld ook lay-outs met veel verschillende kaders gerealiseerd worden. Op deze manier kan men overigens ook zelf een FO renderer ontwikkelen.

3. XML naar PDF en native bestand via opmaaksoftware

Bij de eerste twee opties treedt in de praktijk al gauw een beheersprobleem op als er veelvuldig aanpassingen gewenst zijn van de opmaak: een ander logo, andere kleuren, andere lettertypes, wijziging van positie en formaten van elementen etc. Voor deze wijzigingen is al gauw een programmeur nodig, er is geen mogelijkheid dit interactief aan te passen.

Verder zijn woordafbrekingen dus een probleem en moeten allerlei lastige grafische details zoals kerning (afwijkende spatiëring tussen bepaalde letterparen), overvul (overlappen van kleurelementen), transparantie, afmaskeren van afbeeldingen etc. ontraden (optie 1) of zelf ontwikkeld worden (optie 2).

Voor complexere documenten kan daarom beter gebruik worden gemaakt van 'best of breed' opmaaksoftware waarin deze functionaliteit al jarenlang beschikbaar is. Daarbij kan verschil gemaakt worden tussen DTP software en zwaardere, geautomatiseerde opmaaksystemen die erg kostbaar zijn of minder in zwang zijn bij grafische bedrijven en vormgevers zoals 3B2, XyEnterprise en TeX. DTP software zoals Adobe InDesign, Adobe FrameMaker en Quark Xpress biedt mogelijkheden voor XML import, maar dit is geen geautomatiseerde methode en er zijn beperkingen. Zo is het standaard niet mogelijk om bij aanwezigheid van een bepaald XML element een nieuw kader aan te maken voor tekst of beeld. Er zijn echter voorzieningen waarmee de functies van deze software via een extern script of een commerciële plug-in geautomatiseerd aangeroepen kunnen worden. Wil men dit via een webbrowser activeren dan is het vanwege licentie politieke redenen nodig om server versies van Adobe InDesign of Quark Xpress in te zetten. Voor meer handmatige, single user verwerking kan worden volstaan met de desktop versie van het product in combinatie met een plug-in of aanvullende software.

Het voordeel van deze benadering is dat de beperkingen op grafisch terrein zijn opgeheven; woordafbreking, professionele grafische functies zijn nu direct beschikbaar. Een tweede voordeel is dat deze methode ook ingezet kan worden bij situaties waarbij de opmaak niet 100 % geautomatiseerd kan worden; de PDF fungeert dan uitsluitend als een eerste indicatie van de

opmaak. Vervolgens kan het ‘native’ opmaakbestand door een DTP operator verder geperfectioneerd worden, denk aan het aanpassen van positie, uitsnede en schaling van de afbeeldingen, oplossen van situaties met bijna lege pagina’s etc.

De opmaak gebeurt in dit geval op basis van een template dat door een DTP-operator gemaakt kan worden met het DTP programma zelf (vaak is het ontwerp al als DTP document beschikbaar). Deze kan ook eventuele lay-out aanpassingen zelf uitvoeren, zonder tussenkomst van een programmeur. Installeren en gebruiken van fonts is een gebruikersvriendelijk proces.

De nadelen van deze benadering zijn de hogere licentiekosten (zeker bij web-to-print toepassingen waarbij serverversies van Adobe InDesign of Quark Xpress nodig zijn) en de minder goede performance (in ieder geval bij gebruik van desktopversies). Ook is voor het realiseren van dit soort oplossingen een (vrij schaarse) combinatie nodig van IT en grafische kennis.

Een belangrijk voordeel van deze werkwijze is dat deze ook toepasbaar is als de opmaak niet 100% te automatiseren is. Er wordt in dat geval een halfproduct gegenereerd waarmee een grafisch ontwerper of DTP-er weer verder kan om de puntjes op de (grafische) i te zetten. Zaken als beeldpositie en uitsnede, inwinnen van witruimte en het frivoler maken van de publicatie kunnen dan gewoon handmatig met algemeen toegepaste DTP-software gedaan worden terwijl toch voordelen gehaald worden ten opzichte van een volledig handmatige werkwijze. Zeker binnen de creatieve sector zal deze methodiek favoriet zijn.

De drie mogelijkheden voor XML naar folio omzetting zijn in de volgende tabel samengevat:

	XSL-FO	Software library	DTP + scripting
Opmaak complexiteit	-	+	++
Licentiekosten	++ (FOP) --/- (commercieel product)	+	--/- (server/desktop)
Typering opmaak	tekstflow	Tekstflow + kader	Tekstflow + kader
Implementatie kosten	-/+ (afh. van ervaring)	--	+ (mits ervaring)

Performance	++	++	--/+ (desktop/server)
Resultaat aanpasbaar	-- (PDF) ++ (RTF of native)	--	++
Woordafbreking	-- (FO) ++ (commercieel product)	-	++
Ondersteuning TIFF, EPS, CMYK	-- + (commercieel product)	+	++
Integratie binnen webapplicatie	++	++	+ (via Windows/Mac applicatieserver)
Aanpassen templates	Programmeur (FO) Operator (Commercieel product)	Programmeur, evt. PDF template	DTP-operator
Aanpassen vormgeving variabele gegevens	Programmeur (FO) Operator (Commercieel product)	Programmeur	DTP-operator, soms programmeur

Welke optie de beste is hangt dus sterk van de situatie af, de kosten van de ingezette standaard componenten en de kennis en ervaring bij de bouwers van de oplossing.

Conclusies

XML doet zijn intrede in de creatieve sector, zowel vanwege de ondersteuning in gangbare software toepassingen als vanwege de tendens om in marketing campagnes uitingen via meerdere media en onderling afgestemd te realiseren. Dat betekent niet dat direct iedere creatieve DTP-er of Flash animator ook zelf met XML, DTD's, Schemas, XSLT en FO aan de gang zal gaan. Daarvoor is de XML functionaliteit in de daar gebruikte software nog vrij beperkt. Wel zullen steeds meer programmeurs in de creatieve industrie nodig zijn met kennis van XML, scripting (meestal AppleScript, VBscript of JavaScript) en website ontwikkeling (aanmaak van XML via PHP, ASP, ASP.NET, JSP etc).

Met name Adobe speelt op dit moment een belangrijke rol in deze markt omdat zij toonaangevende producten bieden voor creatie en ontwikkeling van drukwerk, websites, animaties, audio en video en 'rich user interfaces' voor mobiele media.

[1] <http://crossmediaforum.web-log.nl/>

[2] <http://www.dondersteen.nl/>

[3] http://flash-creations.com/notes/dynamic_xml.php

[4] www.openlaszlo.org