

XML naar folio conversie

Screens & Pages whitepaper

Auteur: ir. R.J. Janssen, R.J.Janssen@SandP.nl

Versie: 16/06/2006

Inleiding

De content management markt beweegt zich in de richting van Enterprise Content Management: steeds meer soorten informatie (gestructureerd en ongestructureerd) moeten via meerdere kanalen worden verspreid. Dat betekent dat niet uitsluitend uitvoer voor een website gegenereerd moet worden maar ook voor ‘oude’ media zoals druk- en printwerk en voor nieuwe media zoals smart phones, PDA/pods en e-Books.

Voor gebruikers betekent dit niet veel meer dan een extra menuoptie, maar voor de ontwikkelaars van content management oplossingen ligt dat wat ingewikkelder. Met name de aanmaak van bestanden voor verwerking tot professioneel druk- of printwerk is een bijzondere tak van sport waarvoor vaak specialistische kennis en technologie nodig is. Het kan daarom zinvol zijn om daarbij bedrijven in te schakelen (vaak met roots in de grafische industrie) die in deze materie gespecialiseerd zijn.

Beschikbare opties

Op dit moment zijn er een aantal opties om content automatisch te converteren naar een opgemaakt folio document. 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 een dergelijk XML document te genereren.

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

XSL (eXTensible Stylesheet Language) is een W3C specificatie voor het omzetten van XML naar een andere structuur of tekstformaat. Een bijzonder formaat waarnaar getransformeerd kan worden is XSL-FO, ook een W3C standaard die de opmaak van een document beschrijft in een XML bestand waarvan de structuur voldoet aan een daarvoor ontwikkeld Schema. Zo'n FO bestand kan worden aangemaakt door de bron XML te converteren via een zelf te maken XSL bestand.

Dit FO bestand kan worden aangeboden aan een zogenaamde ‘rendering engine’ die daarvan een opgemaakt bestand maakt dat kan worden opgeslagen als PDF, RTF of ander formaat. Daarbij is de informatie 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 automatisch worden aangebracht.

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

Voor relatief simpele, rapportachtige producten is XSL-FO een snelle, goede en goedkope methode (mits de benodigde kennis beschikbaar is).

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 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).

De programmeur bepaalt nu 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.

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.

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. 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 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.

Het nadeel van deze benadering zijn de hogere licentiekosten (zeker als serverversies nodig zijn) en de minder goede performance (in ieder geval bij gebruik van desktopversies). Ook is voor het realiseren van de oplossing een (vrij schaarse) combinatie nodig van IT en grafische kennis.

Samenvatting

De drie mogelijkheden voor XML naar folio omzetting zijn hieronder samengevat:

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

	ervaring)		
Performance	++	++	--/+ (desktop/server)
Resultaat aanpasbaar	--	--	++
Woordafbreking	--	-	++
Ondersteuning TIFF, EPS, CMYK	--	+	++
Integratie binnen webapplicatie	++	++	+ (via Windows/Mac applicatieserver)
Aanpassen templates	programmeur	Programmeur, evt. PDF template	DTP-operator
Aanpassen vormgeving variabele gegevens	programmeur	programmeur	DTP-operator, soms programmeur
Pricerange eerste publicatietype	5 – 10 k€	8 – 15 k€	Desktop: 12-18 k€ Server: 30-45 k€
Pricerange volgende publicatietypen	4 – 9 k€	2 – 10 k€	Desktop: 2-5 k€ Server: 2-5 k€

Welke optie de beste is hangt dus sterk van de situatie af.

Wat bieden wij

Wij hebben gekozen voor het implementeren van zowel methode 2 als methode 3, zodat we de afweging per situatie kunnen maken. Daarvoor bieden we de volgende standaard tools die we zelf of samen met de implementatiepartner ‘inregelen’ om de juiste opmaak te kunnen genereren:

Methode 2: PDF_ASL

Dit is een .NET applicatie die XML direct omzet naar PDF. Daarbij kan de basisopmaak van de pagina's worden gedefinieerd via één of meer mastertemplates, die eventueel voor linker- en rechterpagina of per deelsectie verschillend kunnen zijn.

De opmaakregels worden toegekend via een XSLT bestand, waarin gegevens staan over te gebruiken paginatemplates, paragraaf en karakterstijlen, kleuren, kaders, tekstflows, tabellen etc.

De toepassing kan worden aangestuurd via een hotfolder en in de nabije toekomst ook als webservice. Integratie met andere systemen en platforms is daarmee goed mogelijk.

Omdat FO zelf ook een XML bestand is kan ook op basis van FO documenten gewerkt worden, bijvoorbeeld als upgrade van een FOP toepassing.

Met deze oplossing kan een hoge performance worden gehaald, typisch enkele 10-tallen pagina's per seconde afhankelijk van de omvang van het beeldmateriaal.

Methode 3: InDesign_ASL

Ook dit is een .NET toepassing die qua interface identiek is aan de PDF_ASL, alleen wordt nu Adobe InDesign CS (Windows desktopversie) of InDesign CS2 (Windows serverversie)

aangestuurd. Afhankelijk van de instellingen kunnen PDF, JPEG en/of Adobe InDesign documenten worden gegenereerd, in zowel lage- als hoge resolutie versies.

De templates worden gemaakt in Adobe InDesign, de opmaakregels worden bepaald via een XSLT bestand.

T.a.v. integratie geldt hetzelfde als bij de PDF_ASL; de performance is lager dan bij de PDF_ASL (van enkele seconden per pagina voor de desktopversie tot enkele pagina's per seconde voor de serverversie). Deze is verder sterk afhankelijk van de gebruikte serverhardware, met name de processor speelt daarbij een grote rol (dualprocessor hardware is aan te raden). Dit is de prijs die moet worden betaald voor compromisloze opmaak en de mogelijkheid om de templates en resultaten door een DTP-operator te kunnen laten aanmaken resp. perfectioneren.

Voor meer informatie:

Roelof Janssen

Screens & Pages

033-4332920

06-54334139

R.J.Janssen@SandP.nl