LinuxFocus article number 242
http://linuxfocus.org

door Floris Lambrechts
<floris/at/linuxfocus.org>

Over de auteur:

Ik ben al een paar jaar de ’beheerder’ van LinuxFocus/Nederlands. Ik studeer voor ’industrieel ingenieur - electronica’ in Leuven. Mijn tijd verspeel ik aan prutsen met Linux, PHP, XML en LinuxFocus, en daarnaast lees ik bv. ook nog boeken (op het moment bijvoorbeeld Stephen Hawking en Jef Raskin’s ’The Human Interface’). En laat ik vooral mijn vriendin niet vergeten (dag Katrien)!

Maak kennis met XML

Kort:

Dit is een korte introductie in XML. Je zult kennismaken met Eddie de meta-kat, de XML syntax politie en DTDs. Geen nood, alles wordt uitgelegd ;-)

Vertaald naar het Nederlands door:
Floris Lambrechts
<floris/at/linuxfocus.org>

Introductie

In de zomer van 2001 kwamen een paar LinuxFocus editors samen in Bordeaux tijdens de LSM. Veel presentaties en discussies in de documentatie-groep aldaar gingen over hetzelfde: XML. Lange, onderhoudende uren kropen in het begrijpen wat XML nu eigenlijk is, wat de voordelen zijn en hoe die te gebruiken. Mocht je geïnteresseerd zijn, dat is ook waar dit artikel over gaat.

Bij deze bedank ik graag Egon Willighagen en Jaime Villate, die me enthousiast hebben gemaakt voor XML. Dit artikel is lichtjes gebaseerd op de informatie in Jaime's artikelen, die je onderaan bij de links kunt terugvinden.

Wat is XML

Wij documentatie-mensen wisten allemaal wel ongeveer wat XML was. Het was toch een taal met een HTML-achtige syntax, meer bepaald een markup taal zoals SGML en (weeral) HTML, niet? Inderdaad. Maar daarmee is niet alles gezegd.
XML heeft een paar eigenschappen die het een nuttig data-formaat maken voor heel uiteenlopende doelen. Het lijkt wel alsof je met XML de meest ingewikkelde dingen kunt, en dat het toch eenvoudig te lezen blijft (voor mensen) en eenvoudig te parsen (voor computers). Hoe is dat mogelijk? We onderzoeken het...

Eddie, de meta kat

Om te beginnen, XML is een *markup taal*. Documenten die je schrijft in een markup taal bevatten twee dingen: *data*, en *metadata*. Als je weet wat 'data' precies is, laat het me weten, maar tot het zover is ga ik het hebben over de metadata ;). Eenvoudig gesteld: metadata is extra informatie die een betekenis, of context, toevoegt aan de data zelf . Een voorbeeld: neem de zin *'Mijn kat heet Eddie'*. Mensen zoals wij weten nu dat *'kat'* een naam is van een diersoort, en dat *'Eddie'* zijn naam is. Computer programma's daarentegen, zijn niet menselijk en weten dit allemaal niet. Nu kunnen we dus metadata gebruiken om betekenis toe te voegen aan de data (uiteraard gebruiken we XML syntax!):

```
<zin>
  Mijn <dier>kat</dier> heet <naam>Eddie</naam>.
</zin>
```

Nu kan zelfs een stom computerprogramma het verschil herkennen tussen de naam van een diersoort (kat) en een eigennaam (Eddie). We kunnen nu bijvoorbeeld een document genereren waarin alle namen blauw zijn, en alle diersoorten rood. XML maakt zo'n omzettingen heel gemakkelijk (het resultaat ziet er zo uit:)

Mijn kat heet Eddie.

In theorie kunnen we alle opmaak-informatie (in dit geval de kleuren) in een apart bestand opslaan, een zogenaamde stylesheet. Al doende hebben we dan de opmaak van de inhoud gescheiden, iets wat door sommigen beschouwd wordt als de Heilige Graal van Webdesign ^{TM}. Maar tot nu toe hebben we eigenlijk nog niet veel bijzonders gedaan. Metadata toevoegen is namelijk iets wat markup talen al jaren kunnen. 'Dus,' dringt de vraag zich op, 'wat maakt XML dan zo speciaal?'

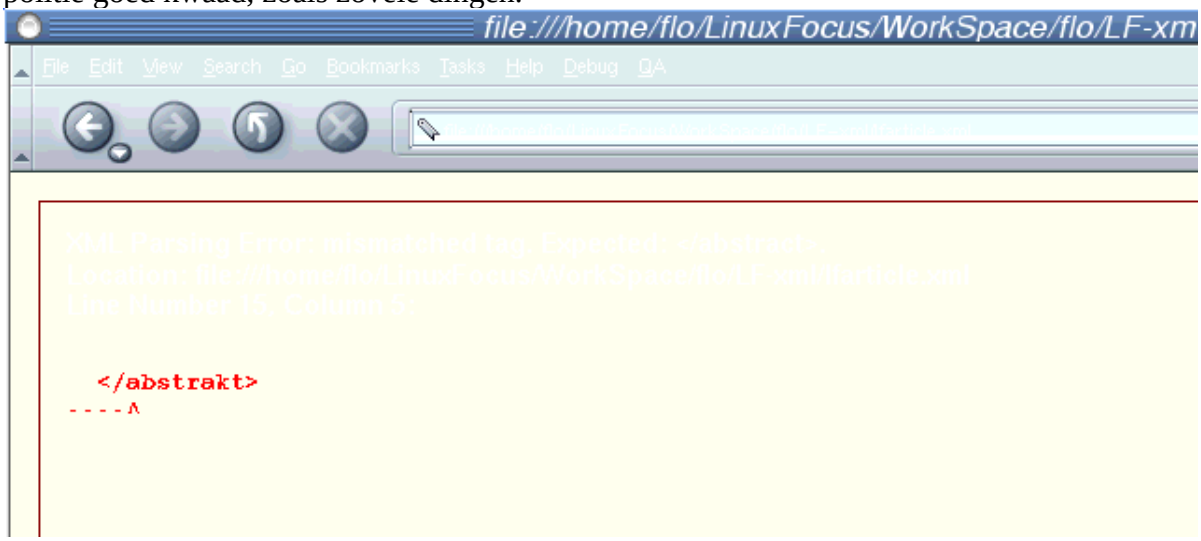
De syntax politie

Om te beginnen heeft XML een heel strenge syntax. Zo moet elke <tag> een afsluitende </tag> heben. [Merk op: omdat het een beetje dom is om twee tags te schrijven <tag></tag> wanneer er niks tussen

staat, kun je de tag ook meteen sluiten wanneer je hem opent, en alzo een paar minuten van je leven winnen, op de lange duur: <tag />.]
Een andere regel is dat je tags niet mag 'mixen'. Je moet ze met andere woorden sluiten in dezelfde (omgekeerde!) volgorde dat je ze hebt geopend. Iets als het volgende is niet geldig:

<B> **Vette tekst** <I>***Vette en schuine tekst*** </B> *schuine tekst* </I>

De syntax regels schrijven voor dat je de </I> tag sluit vóór je de </B> sluit.
En, opgepast, alle elementen in een XML document moeten tussen tags staan (behalve de eerste en de laatste natuurlijk). Daarom hebben we in het voorbeeld hierboven de <zin> tags rond de zin gebruikt. Zonder die tags zouden sommige woorden in de zin niet tussen tags staan, en dat maakt de XML syntax politie goed kwaad, zoals zovele dingen.



Mozilla's syntax politie '@work' ...

Maar zo'n sterke politiemacht heeft natuurlijk zijn voordelen: ze bewaart de orde. De eenvoudige syntax maakt dat programma's heel makkelijk XML kunnen lezen. Bovendien is de data in XML documenten heel gestructureerd, zodat ook mensen het makkelijk kunnen schrijven en lezen.
Merk wel op dat de 'theoretische' voordelen van XML niet altijd opduiken in de praktijk. De huidige XML parsers bijvoorbeeld, zijn meestal verre van snel, en dikwijls heel groot. Dus misschien is XML helemaal niet zo makkelijk te lezen voor computers? Laten we het er gewoon bij houden dat het geen goed idee is om *alles* in XML te doen, gewoon omdat het mogelijk is. Toepassingen die dikwijls documenten moeten doorzoeken, of die heel grote documenten nodig hebben, zijn dikwijls niet zo geschikt voor XML. Maar dat betekent natuurlijk niet dat ze niet realiseerbaar zijn met XML!
Een goed voorbeeld van de kracht van XML, maar ook van de traagheid, is het feit dat je er databases in kunt schrijven (probeer dát maar eens met HTML! :p) Dat is nu net wat Egon Willighagen heeft gedaan voor onze lokale LinuxFocus afdeling, zijn artikel hierover vind je onderaan bij de links. In dit geval verkozen we de flexibiliteit en uitbreidbaarheid van een eigen bestandsformaat boven pure snelheid (bijvoorbeeld met mySQL).
En nog over die strikte syntax: als je goede vriendjes wordt met de politie (de syntax checkers), kun je zelfs wat van je werk op hen afschuiven. Maar in dat geval zul je wel een DTD moeten schrijven...

De DTD

In het ’Eddie de meta-kat’ voorbeeld hierboven hebben we onze eigen XML tags verzonnen. Uiteraard is zo’n daad van creativiteit niet naar de zin van de politie! De ’mannen in’t blauw’ willen weten wat je doet, hoe, wanneer, en (indien mogelijk) waarom. Wel, geen probleem, je kunt alles netjes uitleggen met de DTD...

Een DTD laat je toe om nieuwe tags ’uit te vinden’. Je kan er zelfs een heel nieuwe taal mee ontwerpen, zolang je maar de XML syntax respecteert.
De DTD, of **D**ocument **T**ype **D**efinition, is een bestand met daarin de beschrijving van een XML taal. Eingelijk is het niets meer dan een opsomming van alle mogelijke tags, hun eventuele attributen, en de mogelijke combinaties. De DTD beschrijft wat je kan doen in je XML taal, en wat niet. Dus wanneer we praten over deze of gene ’XML taal’, praten we eigenlijk over een specifieke DTD.

Zet de flikken aan het werk!

Soms zal de DTD je *dwingen* om op een bepaalde plaats een bepaalde tag te schrijven. Sommige tags zijn namelijk verplicht, zoals de begin- en eind-tag, of een tag met de titel van het document. Het leuke hieraan is dat er software bestaat (b.v. een emacs module) die de verplichte tags in jouw plaats schrijft. Op die manier worden sommige delen van je documenten automatisch voor je ingevuld. Omdat de syntax zo strikt en duidelijk is, kan de DTD je als het waren leiden door het schrijfproces. En als je een fout maakt (bijvoorbeeld een tag vergeten te sluiten), dan brengt de politie je op de hoogte. Dus al bij al zijn de flikken nog niet zo slecht: in plaats van te roepen ’Je hebt het recht op stilzwijgen...’, vertelt de XML syntax politie je vriendelijk over een ’Syntax error @ line xx : ’... :)
En als je de politie het vuile werk laat opknappen, heb jijzelf natuurlijk meer tijd om je te concentreren op de inhoud.

In the mix

Een laatste mooie eigenschap van XML is dat je verschillende DTD’s tegelijk kunt gebruiken. Je kunt dus in hetzelfde document verschillende data-types opnemen.

Dit ’mixen’ gebeurt met xml namespaces. Je kunt bijvoorbeeld de DocBook DTD opnemen in je .xml document, in dit geval met de *prefix* ’dbk’.
Al de tags van DocBook staan dan ter beschikking in de volgende vorm: (aangenomen dat er een DocBook tag <gewoon_een_tag> bestaat:)

<dbk:gewoon_een_tag> gewoon wat woorden </dbk:gewoon_een_tag>

Met het namespaces systeem kun je alle tags en alle attributen gebruiken van eender welke DTD. Dit opent een wereld van mogelijkheden, zoals je kunt ontdekken in het volgende hoofdstuk.

Beschikbare DTDs

Hier is een klein overzicht van een paar DTDs die nu al (gedeeltelijk) gebruikt worden.

- **DocBook-XML**

DocBook is een taal om gestructureerde documenten mee te maken, denk aan boeken en papers. Maar je kan het ook inzetten voor andere dingen. DocBook is eigenlijk een SGML DTD (SGML is a markup standaard), maar er is ook een -populaire- XML versie van. Dit is één van de populairste XML DTDs.

- **MathML**
  MathML is de Mathematical Markup Language, gebruikt door wetenschappers om wiskundige symbolen en formules mee te schrijven. Voor mensen uit de wiskundige wereld is dit echt een heel nuttig gereedschap. De chemici van hun kant hoeven niet jaloers te zijn op hun collega's, zij hebben namelijk hun eigen speeltje, te weten CML of Chemical Markup Language. Merk op dat Mozilla 1.0 nu (binnenkort?) standaard MathML ondersteunt.

- **RDF**
  RDF is het Resource Description Framework, ontworpen om metadata te coderen en te hergebruiken. In de praktijk wordt het dikwijls gebruikt door websites om elkaar te vertellen welk nieuws ze tonen. Bijvoorbeeld de Nederlandse site linuxdot.nl.linux.org gebruikt RDF bestanden van andere sites om hun nieuwtjes te kunnen weergeven. De meest populaire nieuws-sites (zoals Slashdot) hebben dan ook een RDF bestand ter beschikking zodat je hun nieuws-headlines b.v. kunt weergeven in een menu op je homepage.

- **SOAP**
  SOAP staat voor Simple Object Access Protocol. Het is een taal waarmee processen met elkaar kunnen communiceren (data uitwisselen en 'remote procedure calls' doen). Met SOAP kunnen twee processen op afstand communiceren, bijvoorbeeld over een http verbinding (internet). Atif van LF zal je hier binnenkort meer over kunnen vertellen :-)

- **SVG**
  Scalable Vector Graphics. Het trio PNG, JPEG2000 en SVG zou de toekomst moeten vormen van afbeeldingen op het web. PNG zal de rol van het huidige GIF overnemen (verliesvrij gecomprimeerde bitmaps met transparantie), en JPEG2000 zal op een dag de .jpg van vandaag vervangen (bitmaps met een instelbare graad van 'verlieslatende' compressie). SVG van zijn kant werkt niet met bitmaps, maar is vector-gebaseerd. Met andere woorden: afbeeldingen worden niet voorgesteld door pixels, maar door wiskundige vormen (lijnen, veelhoeken,...). SVG ondersteunt ook dingen als scripts en animatie, dus het is te vergelijken met Macromedia's Flash. In .svg files kun je JavaScript gebruiken, en met dat JavaScript kan je dan weer .svg code maken. Lekker flexibel he?
  Maar svg is nog nat achter de oren: op dit moment is er enkel een goede SVG browser plugin beschikbaar van Adobe voor Windows en Mac. Mozilla werkt aan een ingebouwde SVG viewer, maar die is nog niet af en je moet een speciale Mozilla-versie binnenhalen om hem aan't werk te zien.
  OPMERKING: .svg bestanden worden algauw heel groot, daarom zie je vaak .svgz bestanden. Die zijn gewoon met het gzip algoritme gecomprimeerd.

- **XHTML**
  XHTML is de XML variant van HTML versie 4.01. De strakke XML syntax veroorzaakt een paar wijzigingen - sommige dingen kan je wel doen in HTML maar niet in XHTML. Maar aan de andere kant is een XHTML pagina wel geldige HTML. Programma's zoals HTML tidy kunnen bestaande HTML pagina's omzetten in XHTML.

- **De rest**
  Veel nieuwe bestandsformaten gebruiken XML, vaak in combinatie met .gz of .zip compressie. Een voorbeeld: de KOffice bestandsformaten zijn XML DTDs. Dit is nuttig, omdat de gebruiker de functionaliteit van 2 toepassingen kan oproepen in 1 document. Je kunt dus een KWord document schrijven met een ingebedde KChart spreadsheet erin.

Links

Het W3C, of World Wide Web Consortium
Ze hebben info over XML, MathML, CML, RDF, SVG, SOAP, XHTML, namespaces...
www.w3.org

Een paar dingen van Jaime Villate (de eerste twee in't Spaans):
Introductie tot XML
Hoe maak je HTML uit XML
LSM-slides

HTML tidy, het programma:
www.w3.org/People/Raggett/tidy

DocBook
www.docbook.org

Mozilla.org SVG project
www.mozilla.org/projects/svg

Relevante LinuxFocus articles:
LinuxFocus.org (/Nederlands) maken met XML en XSLT
PDF documenten maken met DocBook