



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

## Arrivare a conoscere XML



### *About the author:*

Sono stato il capo editore della versione olandese di LinuxFocus per anni. Sto studiando 'ingegneria elettronica industriale' a Leuven, in Belgio, e perdo il mio tempo giocando con Linux, PHP, XML e LinuxFocus e leggendo libri come quelli di Stephen Hawking e (al momento :) di Jef Raskin, 'The Human Interface'.

### *Abstract:*

Questa è una breve introduzione a XML. Incontrerete Eddy il meta-gatto, la polizia della sinsatti XML e qualche DTD. Non preoccupatevi, ci spiegheremo. ;)

---

## Introduzione

Nell'estate del 2001, alcuni editori di LinuxFocus sono venuti a Bordeaux per l'LSM. Molti interventi e discussioni del gruppo specialistico di documentazione dell'LSM si sono rivelati trattare dello stesso argomento: XML. Lunghe (e divertenti) ore sono state spese spiegando cosa sia esattamente XML, a cosa serve e come si può usare. Nel caso foste interessati, questo è anche esattamente l'argomento di questo articolo.

Vorrei ringraziare Egon Willighagen e Jaime Villate per avermi fatto conoscere XML. Questo articolo è più o meno basato sulle informazioni degli articoli di Jaime, che potete trovare tra i link più sotto.

## Cos'è XML

Noi gente che scrive documentazione sappiamo più o meno cosa sia XML. Dopotutto, la sintassi è molto simile a quella dell'HTML ed è solo un altro linguaggio di markup come SGML e (ancora) HTML, giusto? Giusto. Ma c'è di più.

XML ha alcune proprietà che lo rendono un formato utile per praticamente tutto. Sembra quasi che possa descrivere le cose più complesse e comunque rimanere leggibile dagli umani, e facile da analizzare dai programmi. Come può essere? Investighiamo su questo strano linguaggio.

## Eddy, il meta-gatto

Prima di tutto, XML è un *linguaggio di markup*. I documenti scritti in un linguaggio di markup contengono fondamentalmente due cose: *dati* e *metadati*. Se sapete esattamente cosa sono i dati, fatemelo sapere, ma fino ad allora parliamo di meta-dati ;). Detto in modo semplice: i meta-dati sono informazioni extra che aggiungono un contesto, o un significato, ai dati in sè. Un semplice esempio: prendete la frase *'Il mio gatto si chiama Eddy'*. Una persona come voi sa che *'gatto'* è il nome di una specie di animale, e *'Eddy'* è il suo nome. I programmi per computer, d'altra parte, non sono persone e non capiscono queste cose. Quindi usiamo meta-dati per aggiungere un significato ai dati (con la sintassi XML, ovviamente!):

```
<frasee>
  My <animale>gatto</animale> si chiama <name>Eddy</name>.
</frasee>
```

Ora anche uno stupido programma per computer può dire che *'gatto'* è una specie, e che *'Eddy'* è un nome. Se vogliamo produrre un documento dove tutti i nomi sono scritti in **blu**, e tutte le specie in **rosso**, allora XML ce lo rende facile. Giusto per il gusto di vederlo, questo è quello che otterremmo:

Il mio **gatto** si chiama **Eddy**.

Ora, teoricamente, possiamo mettere le informazioni sul layout (i colori in questo caso) in un file a parte, un cosiddetto stylesheet (foglio di stile). Quando lo facciamo stiamo separando le informazioni di formattazione (layout) dal contenuto, qualcosa che è considerato come il Sacro Graal del Web™ da alcuni. Finora non abbiamo fatto niente di speciale, l'aggiunta di meta-dati è il motivo d'essere dei linguaggi di markup. Quindi cosa rende così speciale XML?

## La polizia di sintassi

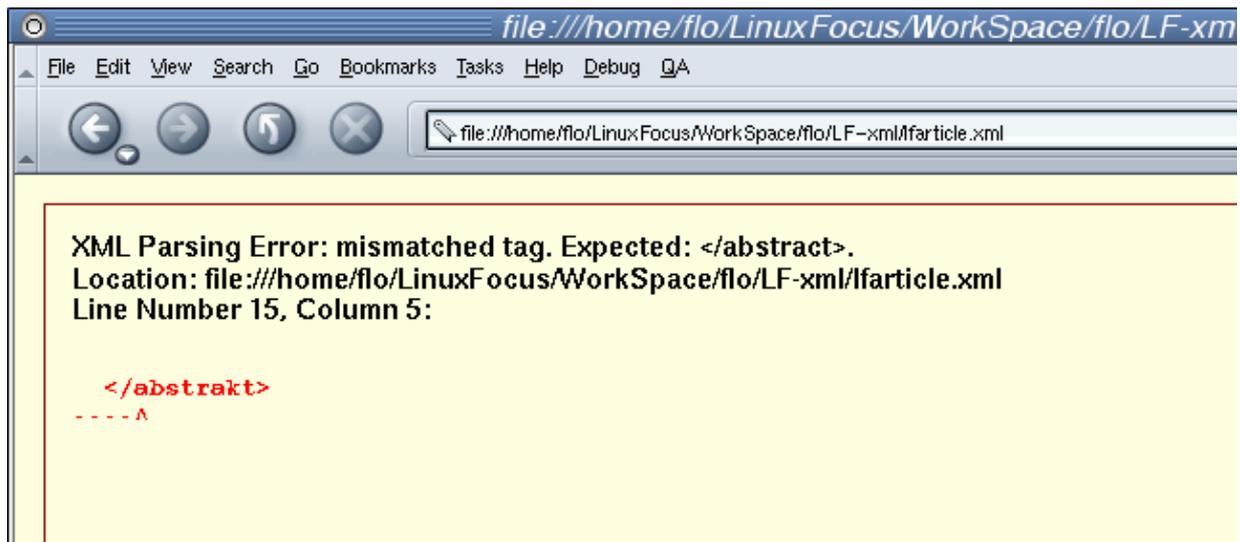
Prima di tutto, XML ha una sintassi molto restrittiva. Per esempio, in XML ogni `<tag>` deve avere un `</tag>` di chiusura. [ Nota: visto che è abbastanza stupido scrivere `<tag></tag>` quando non c'è nulla in mezzo, potete anche scrivere `<tag />` e risparmiare un paio di minuti della vostra vita.].

Un'altra regola è che non potete 'mescolare' i tag. Dovete chiudere i tag nell'ordine inverso rispetto a come li avete aperti. Una cosa come questa non è valida:

```
<B> Testo in grassetto <I> Testo in grassetto e corsivo </B> Testo in corsivo </I>
```

Le regole di sintassi dicono che dovrete chiudere il tag `</I>` prima di chiudere `</B>`

E fate attenzione che tutti gli elementi di un documento XML dovrebbero essere contenuti nei tag (eccetto i due tag esterni, naturalmente!). Questo è il motivo per cui, nell'esempio sopra, abbiamo inserito i tag `<frasee>` attorno alla frase. Senza di essi, alcune delle parole nella frase non sarebbero state racchiuse da tag e questo, assieme a molte altre cose, fa infuriare la polizia della sintassi XML.



### La polizia di sintassi di Mozilla al lavoro...

Ma una forza di polizia con un certo potere ha i suoi vantaggi: assicura l'ordine. Visto che XML segue tali strette regole, è molto facile da leggere da parte dei programmi. Inoltre i dati nei documenti sono molto strutturati, il che li rende facili da leggere da parte delle persone.

In effetti la struttura in XML rende possibile scrivere anche database (provateci con HTML! :p). È esattamente quanto ha fatto Egon Willinghagen per la sezione olandese di LinuxFocus, e il suo articolo su tale sistema è disponibile tra i link in fondo alla pagina.

Se riuscite a diventare buoni amici degli analizzatori sintattici ci sono ancora molti modi di far fare alla polizia il vostro lavoro. Ma per farlo dovete usare in modo intelligente una DTD...

## Le DTD

Nel nostro piccolo esempio 'Eddy il meta-gatto' di sopra, abbiamo inventato i nostri tag XML. Naturalmente un atto così creativo non è tollerato dalle forze di polizia! Gli 'uomini in blu' vogliono sapere cosa stiamo facendo, come, quando e (se possibile) perché. Bene, nessun problema, possiamo spiegare tutto con una DTD...

Una DTD permette di 'inventare' nuovi tag. In effetti permette di inventare un linguaggio completamente nuovo, purché segua la sintassi XML.

La DTD, o **D**ocument **T**ype **D**efinition (Definizione di Tipo di Documento), è un file che contiene la descrizione di un linguaggio XML. È una lista di tutti i possibili tag, dei loro possibili attributi e delle loro possibili combinazioni. La DTD descrive ciò che è possibile nel vostro linguaggio XML, e cosa non lo è. Quindi, quando parliamo di un 'Linguaggio XML', stiamo parlando di una specifica DTD.

### Mettere al lavoro la polizia

A volte la DTD vi *forza* a fare qualcosa in un punto specifico. Per esempio, la DTD può forzarvi a includere un tag che contiene il titolo del documento. La cosa interessante è che ci sono software (per esempio, un modulo di emacs) che scrivono automaticamente i tag richiesti.

In questo modo alcune parti del vostro documento vengono riempite automaticamente. Poiché la sintassi è così stringente e ben definita, la DTD vi può guidare attraverso il processo di creazione del documento. E quando commettete degli errori, come dimenticare un tag di chiusura, la polizia è lì a farvelo notare. Alla fine, questi poliziotti non sono poi così 'furiosi'; invece della frase 'avete il diritto di rimanere in silenzio' della polizia reale, la polizia dell'XML vi dirà amichevolmente 'Errore di sintassi alla linea xx : '... :)

E mentre la polizia fa il lavoro per voi, naturalmente *\*voi\** potete concentrarvi sul contenuto.

## Nel miscuglio

Un'ultima grande feature di XML è la sua capacità di usare diverse DTD allo stesso tempo. Questo significa che potete usare diversi tipi di dati contemporaneamente in un documento.

Questo 'miscuglio' viene fatto con i namespace XML. per esempio, potete includere la DTD di Docbook nel vostro documento .xml (con il *prefisso* 'dbk' in questo esempio).

Tutti i tag di Docbook sono pronti per essere usati nel vostro documento in questa forma (poniamo che ci sia un tag Docbook <solo\_un\_tag>):

```
<dbk:solo_un_tag> solo qualche parola </dbk:solo_un_tag>
```

Usando il sistema dei namespace, potete usare qualsiasi tag e qualsiasi attributo di qualsiasi DTD xml. Questo apre un mondo di possibilità, come vedrete nel prossimo capitolo...

## DTD disponibili

Questa è una piccola collezione di DTD che sono già (parzialmente) in uso.

- **Docbook–XML**

Docbook è un linguaggio per scrivere documenti strutturati, per es. libri e saggi. Ma viene usato anche per lavori molto diversi. Docbook è in verità una DTD SGML (SGML è uno standard di markup), ma c'è anche una versione –popolare– in XML. Questa è una delle DTD più popolari.

- **MathML**

MathML è il Mathematical Markup Language (linguaggio di markup matematico), usato per descrivere espressioni matematiche e formule. È un tool molto utile per le persone del mondo matematico. I chicimi, d'altro canto, non devono essere gelosi dei loro colleghi matematici; anche per loro c'è qualcosa come CML, o Chemical Markup Language (linguaggio di markup per la chimica). Notate che Mozilla 1.0 ha supporto per MathML di default.

- **RDF**

RDF è il Resource Description Framework (Ambiente di Descrizione delle Risorse). È disegnato per aiutare a codificare e riusare i meta–dati; in pratica viene spesso usate dai siti web per dire ad altri siti quali notizie stanno mostrando. Per esempio, il sito olandese [linuxdot.nl/linux.org](http://linuxdot.nl/linux.org) usa i file RDF di altri siti per mostrare le loro notizie. Molti popolari siti di notizie (come per es. Slashdot) hanno un file RDF disponibile per il download, da cui potete copiare i loro titoli da inserire, per es., in una barra laterale del vostro sito.

- **SOAP**

SOAP significa Simple Object Access Protocol (Semplice Protocollo di Accesso ad Oggetti). È un linguaggio usato dai processi per comunicare tra loro (scambiarsi dati e invocare procedure remote). Con SOAP i processi possono comunicare anche remotamente tra di loro, per es. sopra il protocollo HTTP (internet). Penso che Atif qui a LF possa dirvi di più, date un'occhiata ai link. :)

- **SVG**

Scalable Vector Graphics (Grafica Vettoriale Scalabile). Il trio PNG, JPEG2000 e SVG sembra sia il futuro delle immagini sul web. PNG prenderà il posto di GIF (immagini compresse senza perdita di qualità e con trasparenza) e JPEG2000 potrebbe succedere al .jpg di oggi (bitmap con un livello configurabile di perdita di qualità nella compressione). SVG non è un formato bitmap, ma vettoriale, cioè le immagini non sono rappresentate da pixel, ma da figure matematiche (linee, rettangoli, ...). SVG ha anche funzionalità di scripting e animazione, tanto che è comparabile a Macromedia Flash.

Potete usare Javascript nei file .svg, e usando tali Javascript potete anche scrivere codice .svg. Abbastanza flessibile, no?

Ma SVG è abbastanza nuovo; al momento c'è solo un plugin di buona qualità per i browser, disponibile da Adobe per le piattaforme Windows e Mac. Mozilla sta lavorando su un visualizzatore integrato, ma non è ancora completo e dovete scaricare una versione del browser compilata appositamente per usarlo.

*NOTA:* i file .svg possono diventare molto grandi, per cui spesso troverete file .svgz. Queste sono versioni compresse con l'algoritmo gzip.

- **XHTML**

XHTML è la variante XML di HTML versione 4.01. A causa della sintassi stringente di XML, ci sono un paio di differenze – ci sono alcune cose che potete fare in HTML e sono illegali in XHTML. Ma d'altro canto una pagina scritta in XHTML è allo stesso tempo una pagina HTML valida. Notate che il programma HTML tidy può convertire le vostre pagine HTML in XML.

- **Gli altri**

Molti nuovi formati usano XML, spesso combinati con compressione .gz o .zip. Giusto un esempio: Il formato di file di KOffice è una DTD XML. Questo è molto utile perchè permette di combinare le funzionalità di diverse applicazioni in un documento. Per es. potete scrivere un documento con KWrite, con incorporato un foglio di calcolo di KChart.

## Link

Il W3C, o World Wide Web Consortium

Hanno informazioni su XML, MathML, CML, RDF, SVG, SOAP, XHTML, namespace...  
[www.w3.org](http://www.w3.org)

Alcune cose di Jaime Villate (Potreste aver bisogno di un traduttore online per leggere i primi due:)

[Introduzione a XML](#) (in Spagnolo)

[Come generare HTML con XML](#) (in Spagnolo)

[LSM-slides](#)

HTML tidy, il programma:

[www.w3.org/People/Raggett/tidy](http://www.w3.org/People/Raggett/tidy)

Docbook

[www.docbook.org](http://www.docbook.org)

Il progetto SVG di Mozilla

[www.mozilla.org/projects/svg](http://www.mozilla.org/projects/svg)

Articoli correlati di LinuxFocus:

[Usare XML e XSLT per costruire LinuxFocus.org](#) (Olanda)

[Creare documenti PDF con DocBook](#)

---

<p>Webpages maintained by the LinuxFocus Editor team</p>	<p>Translation information: en --&gt; -- : Floris Lambrechts &lt;floris@linuxfocus.org&gt; en --&gt; it: AlessandroPellizzari &lt;alex@at/neko.it&gt;</p>
<p>© Floris Lambrechts</p>	
<p>"some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a></p>	
<p><a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	

