GABRIELE ROMANATO

Internet Explorer 7: storia di un fallimento

Questa traduzione comprende quattro post dal blog di Internet Explorer. Si sono voluti tradurre i post ed i commenti più significativi, al fine di permettere al lettore di comprendere l'evoluzione della release 7 di Internet Explorer.

Sommario

Dettagli delle modifiche CSS in IE7

Attualmente stiamo per lanciare IE7 e vorrei dare un aggiornamento sul lavoro fatto per quanto riguarda i CSS in IE7. Chris ha già delineato i nostri progetti per IE7, e abbiamo ricevuto parecchio feedback (blog, database, conferenze, la nostra partnership con il WASP ecc.) che ci ha aiutato ad affrontare i bug più gravi e a stabilire le priorità per le nuove caratteristiche da inserire in IE7. Vorrei soprattutto ringraziare per la loro partecipazione i commentatori del nostro blog. Il vostro feedback ha fatto la differenza nel decidere quali problemi affrontare.

Siamo consapevoli del fatto di essere ben lontani dalla meta e sappiamo di avere ancora molto lavoro davanti a noi. IE7 rappresenta un passo in avanti nel nostro sforzo di migliorare la nostra conformità agli standard (in particolare per i CSS). Per citare un esempio, nella nostra piattaforma non ci siamo focalizzati su proprietà proprietarie - sebbene potremmo sperimentare nuove caratteristiche in futuro usando il prefisso ufficiale -ms-, seguendo il meccanismo delle estensioni CSS. Abbiamo anche collaborato

con il gruppo di lavoro sui CSS (di cui sono membro) per chiarire i presupposti della nostra implementazione e aggiungere chiarimenti alle specifiche. Vorrei ringraziare tutti quelli che ci hanno aiutato.

Soprattutto, abbiamo apportato oltre 200 modifiche (risoluzione dei bug o nuove caratteristiche) in modalità strict per migliorare la nostra conformità. Tutto questo lavoro (tranne la trasparenza PNG) è stato fatto solo nel the <!DOCTYPE> switch, poiché tutte le modifiche richiedevano degli aggiornamenti per essere al pari delle specifiche CSS. Per mantenere la retrocompatibilità dell'applicazione, non abbiamo apportato modifiche alla modalità quirk stabilita con IE6. Di seguito vi è l'elenco delle nuove caratteristiche e delle modifiche in IE7:

Bug risolti

- Tutti i bug di Position Is Everything tranne l'Escaping Floats Bug (programmato per il futuro)
    - Peekaboo Bug
    - Internet Explorer and Expanding Box Problem
    - Quirky Percentages
    - Line-height bug
    - Border Chaos
    - Disappearing List-Background bug
    - Guillotine Bug
    - Unscrollable Content bug
    - Duplicate Characters Bug
    - IE and Italics
    - Doubled Float-Margin bug
    - Duplicate Indent bug
    - Three pixel text jog
    - Creeping Text bug
    - Missing First letter bug
    - Phantom box bug

Dettagli di altri bug risolti (da risorse diverse da Position Is Everything):

- L'overflow ora funziona correttamente! (I box non si espandono più.)
- Bug nel parser: bug * html, _proprietà e commento /**/
- Controllo select: possibile applicare gli stili
- Dimensionamento automatico degli elementi posizionati con width: auto, right e left (ottimo per layout a tre colonne)
- Risolti molti problemi del posizionamento relativo
- Risolti molti problemi del posizionamento assoluto
- Calcolo dell'altezza/larghezza in % per gli elementi posizionati in modo assoluto http://channel9.msdn.com/ShowPost.aspx?PostID=191182
- Il prologo <?xml> non causa più la modalità quirk
- L'elemento HTML ora realmente indipendente da Body (con una sua larghezza, altezza, ecc.)
- Bordi dotted di 1px non più resi come dashed
- Il bug del margine inferiore su :hover non fa più collassare i margini
- Risolti diversi problemi dei margini negativi
- Problemi risolti nel calcolo del posizionamento relativo e dei margini negativi
- L'attributo CLSID di <object> non è più limitato a 128 caratteri
- Risolto il bug dello spazio bianco su :first-letter descritto in http://blogs.msdn.com/ie/archive/2005/09/02/460115.aspx
- I selettori del discendente ora funzionano correttamente se combinati con altri selettori
- First-line e first-letter ora si applicano quando non c'è uno spazio tra lo pseudo-elemento e la parentesi graffa aperta {
- Le pseudo-classi ora funzionano correttamente se il selettore viene escluso
- Il selettore :link funziona anche per le ancore con href impostato a bookmark
- Risolti i problemi di !important
- piefecta-rigid.html ora funziona
- Risolto il bug dello spazio bianco nelle voci di elenco

- Risolto l'Absolutely Buggy II
- Gli elementi posizionati in modo assoluto ora usano sempre il giusto blocco contenitore per le informazioni di posizionamento e dimensione
- Gli elementi di blocco annidati ora rispettano le dichiarazioni di overflow (hidden, scroll, ecc.)
- Risolto il problema dello scostamento opposto (elementi posizionati in modo assoluto che hanno top, bottom left e right impostate)
- <a> annidati in elementi li non hanno più un margine inferiore extra su :hover
- Il rapporto intrinseco delle immagini non si perde più con il ricaricamento della pagina
- Migliorato il parsing dell'indentazione secondo le regole CSS2.1
- Risolti i bug nel parsing dei selettori di classe multipli combinati con i selettori di id
- E altri ancora

Abbiamo esteso la nostra implementazione per conformarci alle specifiche:

- :hover su tutti gli elementi e non solo su <a>
- Lo sfondo fisso funziona su tutti gli elementi - così il complexspiral demo di Eric Meyer funziona
- Migliorato il supporto ad <object>

Aggiunte nuove caratteristiche dei CSS2.1:

- Supporto a min/max width/height (anche per le immagini)
- Bordi trasparenti
- Supporto al posizionamento fisso
- Selettori: first-child, adiacente, di attributo, del figlio
    - Alcuni selettori di attributo CSS3: prefisso, suffisso e sottostringa a cui stavamo già lavorando (anche il selettore del fratello generico)
- Supporto al canale alfa PNG (non una caratteristica CSS, ma indispensabile ai designers per non far ricorso a JavaScript)

Miglior supporto agli standard...

Come già ricordato, un miglior supporto agli standard in IE significa anche distruggere il layout di alcune pagine. Poichè stiamo lottando per bilanciare i bisogni dei nostri utenti con i desideri degli sviluppatori web, abbiamo bisogno del vostro aiuto. L'unico modo che abbiamo per migliorare il nostro supporto agli standard è poter contare sul vostro aiuto nel cambiare i vostri siti per IE7. Abbiamo fornito un set di documentazione e di strumenti per aiutarvi nella transizione ad IE7:

- Il IE 7 Readiness Toolkit mette insieme documentazione, strumenti, e guide per sviluppatori, tester e professionisti IT per preparare i siti, le estensioni e le applicazioni per IE7.
- La documentazione Cascading Style Sheet Compatibility in Internet Explorer 7 - sulle tecniche da evitare.
- Developer and ITPro Checklists
- Lo IE Developer Center da tutte le informazioni per lo sviluppo (passato e presente).
- Abbiamo un Application Compatibility Toolkit che monitore ed indentifica le modifiche nei comportamenti apportate da IE7 e Vista.
- Tutto questo lo trovate nell'Information Index for IE 7.

Da ultimo, come abbiamo detto, abbiamo una Web Developer Toolbar, che è di grande aiuto nello sviluppo e nel debugging di un sito web.

Stiamo già progettando la prossima release di IE e continueremo sulla strada del supporto ai CSS.

Markus Mielke
Program Manager

Standard e CSS in IE

Sono molto felice del fatto che siamo passati alla versione Beta 1 di IE 7. Volevo precisare che con questa versione abbiamo fatto un piccolo passo

avanti nel miglioramento del nostro supporto agli standard, in particolare dell'implementazione CSS. Non sono invece felice del fatto che non siamo riusciti a completare tutto in questa versione, a causa dei tempi tecnici da dare alla varie release. Tuttavia, so che questo verrà migliorato nella Beta 2 - e voglio condividere con voi quelle che sono le nostre priorità in IE.

Nel team guidato da me, la nostra priorità principale à (e con ogni probabilità sarà sempre) la sicurezza - non un meccanico "evitare la sovrascrittura del buffer", ma qualcosa di innovativo come l'antiphishing e la gestione dei diritti in IE. In particolare, la nostra prossima priorità per IE7 sarà la rimozione di tutto ciò che causa difficoltà agli sviluppatori web. A tal fine, abbiamo studiato molti siti che illustrano i bug di IE, come Position Is Everything e Quirksmode, catalogando ed esaminando questi problemi; abbiamo preso il feedback direttamente da voi (si, noi leggiamo le risposte ai post del nostro blog) a proposito dei bug che vi causano più problemi e su quali nuove caratteristiche vorreste trovare, e abbiamo stabilito ciò che possiamo e non possiamo fare in IE7.

In IE7 risolveremo quanti più bug possiamo, e aggiungeremo anche le più richieste caratteristiche degli standard. Sebbene voi non vedrete tutto questo fino alla Beta 2, abbiamo già risolto i seguenti bug di Position Is Everything e Quirksmode:

- Peekaboo bug
- Guillotine bug
- Duplicate Character bug
- Border Chaos
- No Scroll bug
- 3 Pixel Text Jog
- Magic Creeping Text bug
- Bottom Margin bug on Hover
- Losing the ability to highlight text under the top border
- IE/Win Line-height bug
- Double Float Margin Bug
- Quirky Percentages in IE

- Duplicate indent
- Moving viewport scrollbar outside HTML borders
- 1 px border style
- Disappearing List-background
- width:auto

Inoltre abbiamo aggiunto il supporto alle seguenti caratteristiche:

- Elemento `abbr`
- Migliorato (ma non perfetto) supporto ad `<object>`
- Supporto ai selettori CSS 2.1 (del figlio, adiacente, di attributo, `first-child`, ecc.)
- Posizionamento fisso
- Canale alfa nelle immagini PNG
- `:hover` su tutti gli elementi
- Posizionamento fisso dello sfondo su tutti gli elementi (non solo su `body`)

Vorrei che sia chiaro che il nostro scopo è quello di sviluppare una piattaforma che sia conforme agli standard del Web, in particolare ai CSS 2 (e 2.1, una volta divenuti Raccomandazione). Penso che faremo molti progressi nel far si che in IE 7 vengano rimossi i bug più critici che rendono la nostra piattaforma difficile da usare per gli sviluppatori web.

A questo proposito, ho letto molti commenti che chiedono se supereremo il test Acid 2 pubblicato dal Web Standards Project. Dico subito, eliminando la suspense, che non supereremo questo test al momento del lancio di IE7. L' Acid Test originale testava solo il box model dei CSS 1, divenendo parte della CSS1 Test Suite del W3C, in quanto si tratta di un test abbastanza ristretto. Tuttavia, il test Acid 2 copre un'ampia gamma di standard e di funzionalità (non solo CSS2.1 e HTML 4.01), selezionate dagli autori come una lista di "desideranda" che vorrebbero vedere. Non si tratta di un test di conformità (cito dalla guida al test: «l'Acid 2 non garantisce la conformità con alcuna specifica».) Come tale, è certamente importante ed utile per il mio team, ma non rappresenta, a mio avviso, la nostra priorità per IE7.

Sappiamo bene che IE è in ritardo nel supporto ai CSS. Abbiamo esaminato il test Acid 2 e analizzato i problemi di IE con tale test, e ci siamo assicurati che i bug e le caratteristiche da implementare siano nella nostra lista. Tuttavia, vi sono alcune caratteristiche difficili da implementare, e non le metteremo in cima alla lista delle cose da fare in IE7. Credo che la cosa di gran lunga migliore da fare sia quello di risolvere in primis i bug che vi fanno sbattere la testa sul tavolo e i problemi di usabilità, dando la priorità alla caratteristiche che vengono più richieste in base al feedback che riceviamo.

Sia chiaro: io credo che il Web Standards Project ed il mio team abbiano l'obiettivo comune di semplificare la vita degli sviluppatori web migliorando il supporto agli standard, e sono entusiasta del fatto che stiamo lavorando insieme a tale scopo.

- Chris Wilson

Commenti

re: Standard e CSS in IE

Sabato 30 luglio 2005 11:58 AM di Roberto Scano

ci sarà il supporto per il contenuto application/xhtml+xml (xhtml 1.1) e il supporto per xhtml 1.1 (mappe immagine e notazione ruby)?

Migliorare il parser CSS 2.1 strict in IE 7

Abbiamo già iniziato a parlare di alcuni cambiamenti CSS che saranno disponibili in IE 7 al momento della sua uscita, ma ci sono ancora alcuni punti di cui non abbiamo discusso o che non sono stati affrontati in toto. Ci sono tre punti specifici di cui vorrei parlare:

- Usare il selettore jolly per il nodo radice solo per IE (* HTML) **[risolto solo in modalità strict]**

- Selettori di classe multipli come definito nelle specifiche (.floral.pastel) **[risolto solo in modalità strict]**
- Parsing degli pseudo-elementi che a volte genera regole non valide (P:first-letter{ color: red; }) **[risolto in modalità strict/quirks]**

Selezione del nodo radice:

Per essere chiari, il selettore del nodo radice era un bug. Fu introdotto da Chris Wilson in IE 4, ed è per questo motivo che non gli permettiamo più di lavorare con il codice.

Il selettore del nodo radice è stato a lungo usato per creare regole funzionanti solo in IE. Lo schema generale sarebbe quello di creare regole che funzionino in primis in tutti i browser. Quindi si può usare il selettore del figlio (>) per creare regole più specifiche che funzionino solo in quei browser che supportano tali selettori. Prima di IE 7 non supportavamo questi selettori, tanto che questi erano invisibili ai nostri occhi. Infine si potrebbe usare il pattern * HTML per selezionare il nodo radice. Poiché gli altri browser non supportano o contengono un nodo del DOM situato sopra il nodo HTML, questi pattern non avranno efficacia.

Ma cosa succede quando si cominciano ad usare i selettori del figlio in IE 7? Bé, questo crea una grossa confusione, in quanto ora andiamo a selezionare regole che non sono concepite per essere usate su IE e queste ultime si mescolano con le regole specifiche per IE. A causa della confusione degli stili si finisce per modificare il layout della pagina voluto dall'autore, concepito per avere una compatibilità cross-browser.

La soluzione migliore in questo caso consiste nel disabilitare la selezione dell'elemento radice, in quanto si presuppone che essa non funzioni secondo gli standard. Facciamo questo solo in modalità strict, in quanto i nuovi selettori funzionano solo in questa modalità, ed in questo modo usiamo lo stesso insieme di regole usate dai browser conformi, ignorando le regole specifiche per IE di una pagina rappresentata secondo il design dell'autore. Ci sono alcuni problemi, specie quando la nostra interpretazione

delle specifiche differisce da quella delle altre implementazioni, ma nella maggior parte dei casi si tratta di problemi minori che possono essere agevolmente aggirati.

Un articolo su Peachpit di Molly Holzschlag contiene ulteriori esempi sul funzionamento di quanto detto sinora.

Selettori di classe multipli:

Quando fu sviluppato il supporto per il selettore di classe ci si basò sulle specifiche CSS 1, che supportavano solo un singolo selettore di classe in ogni selettore semplice. Continuammo a mantenere questo comportamento anche dopo aver implementato le successive versioni delle specifiche. Il risultato finale è che le classi extra nel selettore vengono ignorate, mantenendo solo l'ultima della lista e basando la selezione su quest'ultima.

Bé, alcuni siti usano i selettori di classe multipli, così quando ci dedicammo al supporto ai selettori CSS 2.1 fu abbastanza semplice aggiornare i nostri selettori di classe in modo da permettere di applicare più di una classe. Quando si è in modalità strict vengono ora rispettati tutti i selettori di classe specificati per ciascun selettore semplice. Per quanto questa caratteristica non sia usata spesso, la si può utilizzare per alcune applicazioni interessanti. Uno dei miei test originali usava combinazioni di classi rosse, gialle e blu per rappresentare gli elementi in base alla combinazione di colori. Un selettore come .red.yellow rappresenterebbe l'elemento arancione, se entrambe le classi sono contenute un elenco separato da spazi all'interno dell'attributo di classe. Gli altri elementi non verrebbero selezionati, ed in questo modo si possono applicare stili gerarchici più accurati.

DHTML Kitchen possiede un grande esempio di selettori di classe multipli e gli attuali problemi di compatibilità.

Selettori di pseudo-elementi non validi:

Abbiamo applicato un'interpretazione assai rigorosa degli pseudo-elementi nel nostro parser, e questo farà in modo che alcuni costrutti vengano ignorati. Fondamentalmente crediamo che ogni pseudo-elemento debba essere davvero l'ultima parte di un selettore. Le specifiche affermano che può esservi solo uno pseudo-elemento per selettore e che debba comparire all'interno dell'ultimo selettore semplice all'interno del selettore. A causa di questa interpretazione rigorosa, se qualsiasi carattere diverso da uno spazio bianco o un token dovesse comparire dopo l'elaborazione di uno pseudo-elemento, verrebbe associato un flag di errore alla regola. Questo ci da il seguente comportamento:

- Fallisce - `P:first-letter{ color: blue; }`
- Fallisce - `P:first-letter:hover { color: blue; }`
- Funziona - `P:first-letter { color: blue; } /* notare lo spazio */`
- Funziona - `P:hover:first-letter { color: blue; } /* notare l'ordine */`

Il parser è molto più intelligente su quando e come applicare il flag di errore in IE 7, e i due fallimenti visti ora funzioneranno. Le regole realmente scorrette falliranno ancora, e dovrete prestare attenzione a non applicare pseudo-elementi multipli o pseudo-elementi che si trovano in selettori semplici all'inizio di un selettore complesso.

Per un chiaro esempio del problema potete leggere l'articolo su MaxGeek.com.

Andando avanti:

Questi piccoli problemi, che rendevano la scrittura delle pagine secondo gli standard una situazione da continui esperimenti, sono stati risolti in IE 7. Migliorando la logica del parsing diventa più ovvio sapere come i vostri selettori debbano essere scritti, e la documentazione esistente del W3C può essere usata per velocizzare il tutto. Dovrebbe essere facile creare layout interessanti e formattare le vostre pagine senza dover specificare

regole per ciascun browser e speriamo che IE 7 faccia un passo avanti per renderlo reale.

- Justin Rogers

Commenti

re: Migliorare il parser CSS 2.1 strict in IE 7

Sabato 3 settembre 2005 7:43 AM di Ian Hickson

Attenzione! La soluzione da te descritta per gli pseudo-elementi non è del tutto esatta. Il seguente esempio **dovrebbe fallire**:

```
p:first-letter:hover
```

La regola è che uno pseudo-elemento può essere aggiunto all'ultimo selettore semplice in una concatenazione, nel qual caso l'informazione di stile si applica ad una sottoparte di ciascun soggetto. Così i seguenti esempi sono validi:

```
P:first-letter,DIV{}
P:first-letter,DIV:first-letter{}
P.foo:hover:first-letter{}
```

... ma i seguenti NON sono validi:

```
P:first-letter DIV{}
P:first-letter:first-letter{}
P:hover:first-letter.foo{}
P.foo:first-letter:hover{}
P:first-letter.foo:hover{}
```

Tutti quelli dell'ultimo elenco dovrebbero far si che l'intera regola sia ignorata. Mandami un'email se hai bisogno di più informazioni o test: ieblog@hixie.ch

re: Migliorare il parser CSS 2.1 strict in IE 7

Sabato 3 settembre 2005 6:40 PM di Ingo Chao

Justin, mentre stavate fixando il parser, avete fixato anche questo?

```
p {color: yellow;}
p#u.a {color: red;}
p#u.b {color: blue;}

<p id="u" class="b">
Questo dovrebbe essere blu, ma IE6 fallisce con
p#u.a, e non guarda a p#u.b per qualche motivo.
Quindi non è blu, non è rosso, ma giallo.
.</p>
```

re: Migliorare il parser CSS 2.1 strict in IE 7

Sabato 3 settembre 2005 6:59 PM di Ingo Chao

E che dire del divertimento con gli ID e gli pseudo-elementi?

```
p#nonexistantid a:first-letter {color: yellow;}
p#u a:hover {color: red;}

<p id="u">
<a href="#">
Questo dovrebbe essere rosso su hover</a>
</p>
```

re: guardando indietro con nostalgia

Domenica 4 settembre 2005 3:31 AM di Ingo Chao

Justin, un'ultima domanda su questi "bug da farti sbattere la testa sul tavolo" [Chris Wilson, 29 luglio]:

```
a {display: block;}
p#nonexistantid a:first-line {color: yellow;}
p#u a:hover {color: red;}

<p id="u">
```

```
<a href="#">Puoi dirmi per favore
<br />perché solo la prima riga
<br />reagisce su hover?</a>
</p>
```

Il giorno in cui tutti questi bug saranno spariti (e nessun altro problema verrà costruito, vedi gli avvertimenti nei commenti di Ian Hickson e Lachlan Hunt), non penso che guarderò indietro con nostalgia, ma credo di che mi sarò in qualche modo abituato ad essi a causa degli "hack" **così** soprendentemente intuitivi:

```
a:hover img {border: 1px solid fuchsia; }
/* un non-commento per fixarlo */
/* a:hover {background-position: 0 0; }*/

<p><a href="#">
<img src="#" alt="hover" /></a></p>
```

Se IE si unisse agli altri browser che de facto permettono di costruire layout che "funzionano fuori dal box", quello sarebbe un giorno radioso.

Ci sarebbe più tempo per cose serie come il giardinaggio, e meno tempo perso nell'assurda risoluzione dei bug dei browser.

Il prologo <?xml>, la modalità strict e l'XHTML in IE

Leggendo i commenti all'ultimo mio post ho capito di aver dimenticato di citare una voce importante nella mia presentazione. Abbiamo sistemato il DOCTYPE switch in modo da saltare il prologo XML, affinché l'XHTML valido venga trattato in modalità strict piuttosto che quirk.

Ho anche letto dei commenti nel blog di IE che chiedevano il supporto per il MIME type "application/xml+xhtml" in Internet Explorer. Devo dire che IE7 non estenderà il supporto a questo MIME type - continueremo, naturalmente, a leggere l'XHTML quando questo viene servito come "text/html", presupponendo che segua le raccomandazioni di compatibilità

dell'HTML. Abbiamo risolto direttamente i problemi con il nostro DOCTYPE switch in modo tale che ora questo meccanismo è più facile da usare, ed è generalmente facile impostare i server in modo condizionale per servire il contenuto come "text/html" quando il MIME type "application/xml+xhtml" non è supportato.

Per quale motivo non supporteremo l'XHTML quando viene servito con il MIME type "application/xml+xhtml" in IE7? Ho preso la decisione di non supportare il MIME type in IE7 semplicemente perché, personalmente, vorrei che l'XHTML abbia successo sulla lunga distanza. Amo l'XHTML (guardate, il mio nome è nei ringraziamenti per XML 1.0); può davvero essere interoperabile se fatto correttamente. Con la maggior parte delle risorse per IE7 (a parte il lavoro svolto per la sicurezza) spese per migliorare il nostro supporto CSS, se avessimo cercato di supportare il vero XHTML in IE7 avremmo finito per usare il nostro parser HTML (che si concentra sulla compatibilità) e hackerare i costrutti XML. È molto improbabile riuscire a supportare l'XHTML in questo modo; in particolare, non riusciremmo certamente ad individuare alcuni casi di errore e supporteremmo implicitamente casi non validi. Naturalmente questo causerà in futuro problemi di compatibilità basati sulla gestione degli errori nel parser, cosa che l'XML cerca esplicitamente di evitare; non vogliamo causare una confusione simile all'attuale gestione degli errori nell'HTML (che affonda le sue radici nella compatibilità con i vecchi browser - potete biasimare me per questo, ma non IE). Vorrei davvero avere il tempo di implementare correttamente l'XHTML dopo IE7, e fare in modo che sia realmente interoperabile - ma era nostra intenzione sbloccare lo sviluppo dell'XHTML facendo del nostro meglio, ed ecco perché abbiamo affrontato il problema del prologo XML e del DOCTYPE.

- Chris Wilson

Commenti

re: Il prologo <?xml>, la modalità strict e l'XHTML in IE

Giovedì 15 settembre 2005 5:08 PM di Dean Edwards

Chris, un commento SGML causerà anch'esso il quirks mode se viene incluso prima del DOCTYPE. Avete risolto anche questo?

re: Il prologo <?xml>, la modalità strict e l'XHTML in IE

Venerdì 16 settembre 2005 10:18 AM di Henri Sivonen

Grazie per non aver aggiunto il "supporto" ad application/xhtml+xml hackerando il codice HTML. Il Web verrà servito in modo migliore da una corretta implementazione, anche se questa richiede più tempo.

I commentatori che pensano che un corretto supporto XHTML sia solo questione di usare un altro parser dovrebbero sapere che le modifiche che richiedono più lavoro sono quelle che fanno usare al layout il DOM XML. Ci sono sottili differenze:
http://www.mozilla.org/docs/web-developer/faq.html#xhtmldiff