

Ricette per sviluppare applicazioni Web



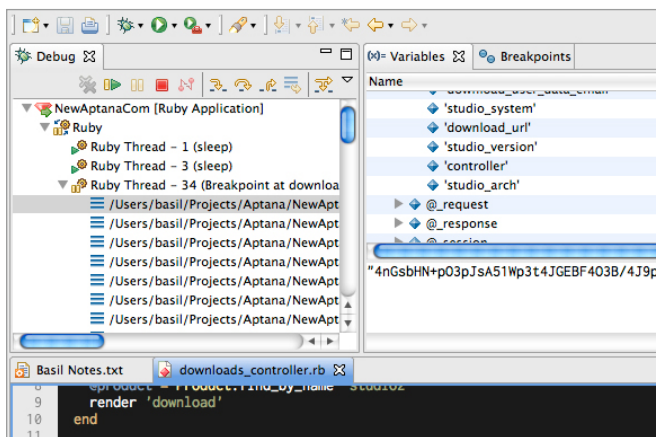
Cms o programmazione su misura? Quale strumento scegliere per disporre di un ambiente adeguato alle finalità del proprio sito.

Nel numero precedente della rubrica abbiamo descritto brevemente le procedure e ambienti grafici Open Source più interessanti per lo sviluppo di applicazioni *da desktop*. Questo mese faremo la stessa cosa per lo sviluppo Web, che per vari motivi ha caratteristiche profondamente diverse. Prima di tutto, la maturità e flessibilità dei sistemi di gestione dei contenuti (*Content Management System* o *Cms*, spesso Open Source come WordPress o Drupal) ha reso inutile per alcune applicazioni "sviluppare per il Web" in senso stretto, cioè scrivere più o meno da zero codice sorgente per un qualche portale o servizio online. Basta conoscere a fondo le caratteristiche del *Cms* scelto, compito a volte tanto complicato quanto programmare ma di tutt'altra natura, per fare tutto quello che serve. Ci sono però tanti altri casi, oltre a quello in cui il software che serve è proprio un *Cms*, in cui scrivere codice

rimane inevitabile. Rientrano in questa categoria i siti in cui l'aspetto estetico e l'interazione avanzata con l'utente sono requisiti essenziali e quelli in cui bisogna svolgere calcoli molto pesanti o utilizzare librerie particolari, non integrate nei *Cms* normali. In situazioni del genere configurare e modificare quei programmi, col rischio di dover ripartire da zero a ogni loro aggiornamento, potrebbe essere più complicato della realizzazione di un'applicazione dedicata.

Quando questo avviene essere già un programmatore esperto aiuta, ma meno di quanto si potrebbe pensare. Un'applicazione o servizio che vive all'interno di

un browser, dai blog all'home banking e ai giochi online, ha vincoli molto diversi da quelli che si dovrebbero rispettare con software nativo per questo o quel sistema operativo. Questa differenza di fondo si ripercuote anche sulle relative procedure e ambienti di sviluppo. La creazione di interfacce utente con modalità grafiche, tanto per fare un esempio, trascinando pulsanti, menu e altri elementi nelle posizioni desiderate è meno sofisticata (quando è possibile) che negli ambienti di sviluppo integrati per desktop. Librerie come jQuery (vedi il box in queste pagine) permettono di aggiungere un menu a tendina in un sito in pochissimo tempo, ma per farlo

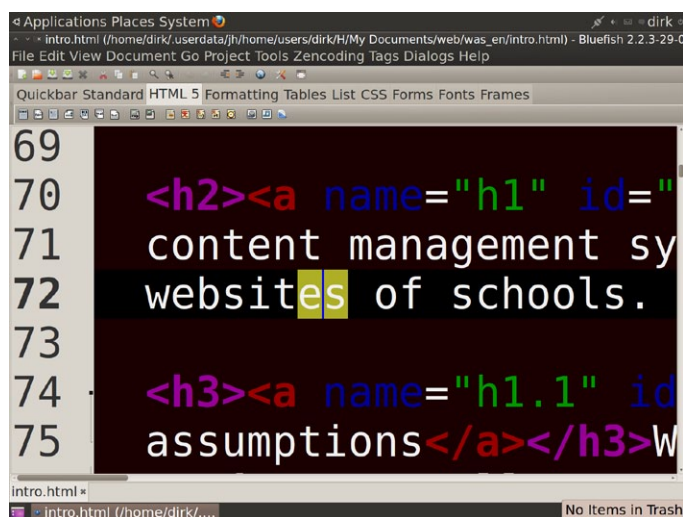


Il debugger di Aptana Studio aggiunge breakpoint per analizzare come varia lo stato di un'applicazione Web JavaScript o Ruby On Rails a ogni istruzione.

I vantaggi dell' Html 5?

L'*HyperText Markup Language*, Html per brevità, è il linguaggio di base di tutti i siti Web esistenti. È Html che specifica come ogni parte di una pagina Web deve apparire in un browser, qual è la sua funzione, come inserire immagini e soprattutto i link ad altre pagine. La versione 5 di questo linguaggio, nato più di vent'anni fa, supporta nuovi marcatori (*tag*) all'interno delle pagine e nuove funzioni interattive codificate in JavaScript. Questo significa (almeno in teoria) sistemi finalmente nativi e interoperabili per integrare video e audio nelle pagine, indicizzazione dei contenuti più facile per i motori di ricerca. Altre novità sono inserimento della geolocalizzazione, dovuta all'incremento dell'utilizzo dei dispositivi mobili, e aggiunta del Web Storage che andrà a sostituire gli attuali cookie.

L'editor di Bluefish è adatto anche a utenti ipovedenti: schemi di colori, dimensioni e comportamento del cursore sono altamente programmabili.



si deve comunque scrivere codice in un editor di testo, sapendo esattamente *dove* metterlo e come verrà influenzato dalla struttura della pagina Html che lo contiene o dal suo foglio di stile Css.

Siti Web d'autore

Per controllare ogni minimo particolare dell'aspetto grafico e della struttura di un sito si può usare un Ide (*Integrated Development Environment*, cioè ambiente di sviluppo integrato). Oggi gli Ide Open Source sono in numero minore di qualche anno fa ma anche, in generale, più maturi e stabili, oltre a girare anche su OS X e Windows.

Aptana Studio (www.aptana.com), per esempio, funziona sia da solo sia come plugin dell'ambiente più complesso chiamato Eclipse. Tutti i principali linguaggi e standard di programmazione, da Css3 e JavaScript ad Html5 e Php sono immediatamente utilizzabili. Altri, come Ruby on Rails, PHP, Python, Perl, Adobe AIR o Objective-C per il mondo iPhone-iPad, richiedono plugin separati.

L'editor di Aptana offre completamento automatico delle parole chiave di codice e, soprattutto, documentazione in linea per Php e altri linguaggi: basta passare con il mouse su una funzione predefinita e premere il tasto Control per aprire una finestra sulla sua definizione completa. Il debugger integrato aiuta a individuare più rapidamente possibile gli errori, tramite varie funzioni per eseguire il codice a piccoli passi, in maniera controllata. L'interfaccia grafica include un terminale per eseguire velocemente script o comandi di sistema ogni volta che è necessario.

Le varie versioni di ogni progetto sono conservate localmente, agganciandosi al sistema di controllo versioni Git, descritto nell'altro articolo su questo numero. Una volta pronto, l'intero sito può essere caricato sul server che lo ospiterà con un solo comando e poi tenuto costantemente in sincronia con le modifiche fatte sul proprio computer.

Un'altra Ide per Web Open Source Molto è Bluefish (<http://bluefish.openoffice.nl/>): a differenza di Aptana Studi, per sfruttarlo appieno occorre servirsene su Linux, poiché non tutte le funzioni sono già disponibili per gli altri sistemi operativi. L'interfaccia grafica è relativamente semplice ma potente, con modelli e wizard anche per codice C, Docbook, Sql e configurazione di server Apache. In Bluefish è possibile (ammettendo che sia *necessario*, ma quello è un altro discorso) aprire centinaia di file simultaneamente, anche in maniera automatica. Si possono caricare con un solo comando tutti i file con una certa estensione presenti in una cartella e in tutte le sue sottocartelle, anche su computer remoti se si ha una connessione adeguata. A molti utenti, comunque, capiterà molto più spesso di elaborare così tanti file con

RISORSE

Gli ambienti di sviluppo Web descritti hanno tutti una buona documentazione accessibile in linea, o dalle rispettive home page. Il punto di partenza (semi) ufficiale per conoscere meglio jQuery è la pagina www.w3schools.com/jquery/jquery_intro.asp. Le linee guida per sviluppare applicazioni Html5 Ubuntu, complete di esempi e link ai vari componenti software necessari e alla documentazione, si trovano su <http://design.ubuntu.com>. La galleria all'indirizzo www.smartclient.com/smartgwt/showcase/, invece, contiene centinaia di esempi d'uso di SmartGWT.

Bluefish solo indirettamente, per esempio chiedendogli di modificare tutti i file di un progetto con il suo potente motore di ricerca e sostituzione testo. Questa categoria di funzioni Bluefish include la sostituzione di tab con spazi, la rimozione di spazi in eccesso e altre operazioni che, se non sempre rendono più leggibili i file Html, possono ridurre sensibilmente le dimensioni.

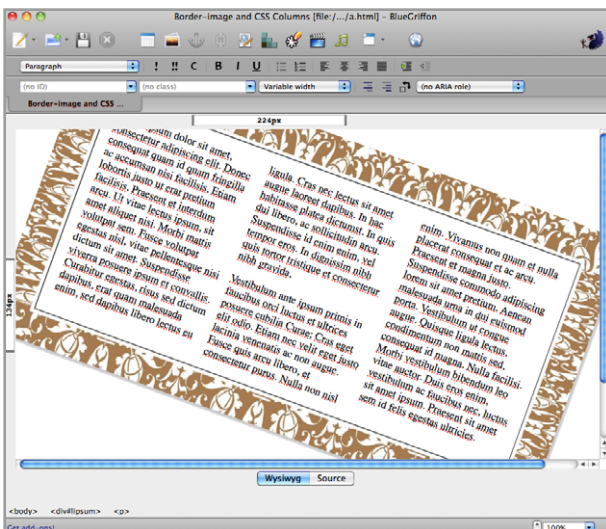
L'editor riconosce, colora e spesso completa automaticamente, senza problemi, anche diversi linguaggi incastriati l'uno nell'altro nello stesso file, cosa comunissima, anzi inevitabile, in qualsiasi pagina Web dinamica. Molti elementi Html a contenuto variabile, come i campi a scelta multipla di un modulo, si riempiono velocemente grazie ai wizard che Bluefish lancia quando si fa clic sui pulsanti o voci di menu corrispondenti. Lo stesso discorso vale per le immagini, di cui si possono specificare bordi, didascalia, testo alternativo ed eventuale anteprima in un solo pannello.

L'altro punto di forza di questo programma è la facilità con cui si possono integrare nell'interfaccia grafica, associandole a pulsanti o combinazioni di tasti veloci, varie utility da riga di comando. Qualsiasi distribuzione Linux

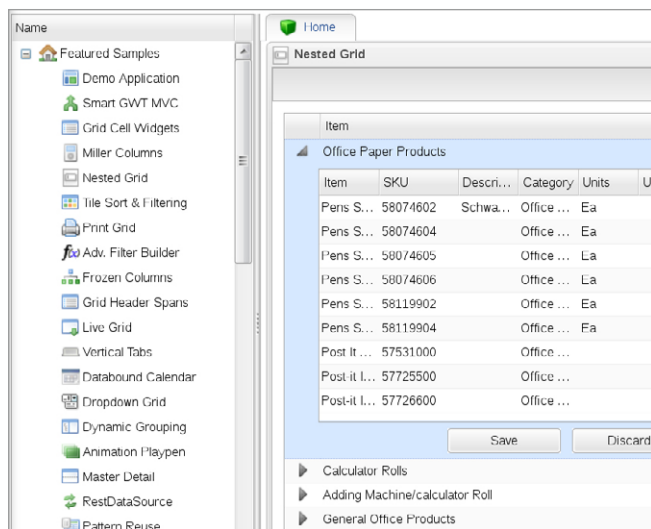
Non dimentichiamo jQuery!

Molte funzionalità dei siti Web sono applicazioni di una piccola porzione di codice Open Source: la libreria JavaScript chiamata jQuery (<http://jquery.com>). Compatibile con i maggiori browser e racchiude la maggior parte delle funzioni di interazione. Dai menu a tendina all'aggiornamento in background di *alcune* parti della pagina, con jQuery tutto si risolve in chiamate a funzioni predefinite, semplici e rapide.





Con BlueGriffon impostare stile e layout grafici delle pagine Web è talmente facile che si rischia di esagerare.



Le librerie per l'ambiente di sviluppo Gwt contengono gli elementi necessari per sviluppare applicazioni Web molto sofisticate.

è piena di programmi del genere, dal "make" descritto nel numero precedente ai vari strumenti che analizzano o ripuliscono automaticamente il codice Html. Lint, Weblint e tidy sono solo tre delle alternative disponibili su Linux e integrabili in Bluefish, per eliminare tutti i marcatori non standard e segnalare stili di codifica da evitare. I pulsanti usati più frequentemente si possono disporre in una barra degli strumenti speciale, chiamata *Quickbar*.

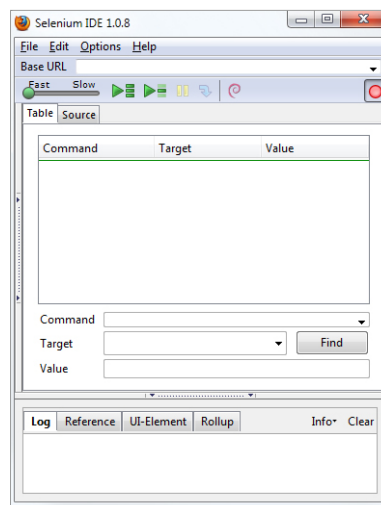
BlueGriffon

Questa potrebbe essere la scelta migliore per chi ha intenzione di sfruttare tutte le potenzialità della versione 5 del linguaggio Html. Anche negli altri casi, ovvero quando Html5 non serve, questo software è una buona soluzione per chiunque voglia scrivere o aggiornare pagine Web (comprese quelle inizialmente create con altri programmi) rispettando i più recenti standard aperti del Web, ma senza doversi studiare a fondo. (<http://bluegriffon.org/>)

In pratica, lavorare con BlueGriffon potrebbe richiedere un po' di tempo per ambientarsi ma è un buon investimento se si vogliono realizzare pagine efficienti, piacevoli e compatibili con tutti i browser. Un apposito menu a tendina contiene tutti gli elementi Html5, mentre finestre di dialogo separate facilitano la configurazione degli elementi di un modulo. La scelta e l'applicazione dei font Open Source di Google, o di quelli del portale FontSquirrel (www.fontsquirrel.com) avvengono in un'apposita scheda, senza scrivere alcun codice.

Bastano pochi clic per estrarre fotogrammi da un video e usarli come immagini o insegne di un sito. C'è un editor integrato per disegnare e incorporare grafici vettoriali nei documenti e se questi ultimi sono in Html4, che non supporta nativamente quei formati, nessun problema! BlueGriffon inserirà automaticamente il codice JavaScript per far caricare ai browser degli utenti i plugin necessari.

Le tabelle (quando servono davvero e non si dovrebbero invece usare i fogli di stile) sono dimensionabili con



L'interfaccia grafica di SeleniumHQ, l'applicazione Open Source per effettuare test automatici tramite browser.

grande accuratezza tramite i righelli orizzontale e verticale dell'interfaccia grafica. Come per i moduli, anche qui per aggiungere righe o colonne o altri elementi, basta un clic.

Passando ai fogli di stile delle pagine Web, anche qui non serve conoscerne lo standard *Css* (*Cascading Style Sheet*) per servirsene. Ci pensa l'apposita scheda "Proprietà di stile" ad automatizzare l'applicazione dello stile prescelto a tutte le pagine di un sito. Altre funzioni di BlueGriffon, più o meno invisibili ai suoi utenti, si occupano di creare le varianti di uno stesso foglio necessarie per diversi browser o diversi media (schermi o stampa). Paradossalmente, si potrebbe dire che il maggior problema dell'uso di *Css* in BlueGriffon è che è talmente semplice che è facile strafare. Per ottenere i bordi grafici e il testo inclinato della figura in alto a sinistra, tanto sofisticati tecnicamente quanto poco pratici e di dubbio valore estetico, bastano infatti pochi clic.

L'ultima cosa da dire su BlueGriffon è che le sue già notevoli capacità si possono notevolmente aumentare con decine di plugin, anche se molti di questi ultimi sono a pagamento. La scelta va da un editor *Css* a uno per formule (compatibile con il macro *LaTeX*) e a un *Mobile Viewer*. Quest'ultimo mostra in anteprima come apparirà il sito che si sta creando in uno qualsiasi delle migliaia di smartphone e tablet presenti nel suo database.

File sempre sotto controllo

In tutti gli ambienti di sviluppo per il Web sono presenti, i cosiddetti sistemi di controllo delle versioni (Scv). I sistemi Scv non devono essere mai confusi con le soluzioni per il backup, perché hanno uno scopo molto diverso. Un programma di backup crea copie complete, in un certo momento, di tutti i file che interessano, in cartelle esterne al computer o alla rete locale: il suo scopo è garantire agli utenti che potranno recuperare tutti i loro file, in caso di "catastrofi" come guasto o furto del computer o semplici cancellazioni erronee.

I sistemi Scv servono invece per conservare tutte le versioni di ciascun file, per poter sapere in ogni momento chi ha effettuato ognuna delle sue modifiche, quando e possibilmente anche perché. L'altro loro uso principale è consentire di tornare in ogni istante, automaticamente, a usare una precisa combinazione di versioni dei file. È grazie a software di questo genere che programmatori di tutto il mondo riescono a collaborare senza perdere la testa, anche se non si conoscono e non si incontrano mai, su programmi da milioni di righe di codice come il kernel Linux.

Esistono due categorie di sistemi di controllo: quelli centralizzati, o client-server, e quelli distribuiti (p2p). Entrambi fanno lo stesso lavoro, ma la prima categoria crea e custodisce una copia "ufficiale" (repository) di tutti i file di un progetto su un unico server centrale. Quando un membro del progetto deve modificare uno o più file, per prima cosa dichiara al sistema Scv su quale versione o combinazioni di versioni desidera lavorare. Questo gli dà la sicurezza di "vedere" nel proprio computer solo quelle versioni, anche se ne esistono altre decine o centinaia. Una volta effettuate e validate le modifiche, il programmatore dovrà caricarle e registrarle (check-in) presso il server, sempre con procedure gestite dal software Scv per evitare confusioni e conflitti con altri membri del gruppo. Solo a quel punto gli altri membri del gruppo vedranno (se lo desiderano) quelle nuove versioni. **Nei sistemi Scv distribuiti** manca l'archivio centralizzato ufficiale: ognuno dei partecipanti ha una copia completa di tutti i file del progetto, o almeno della parte che gli interessa, e il software Scv provvede a tenere tutte le copie in sincronia fra loro.

Almeno in prima approssimazione, tutti i sistemi Scv funzionano allo stesso modo, a parte i differenti nomi dei comandi e delle rispettive opzioni. Dopo aver preso confidenza con uno qualunque di loro, passare a uno degli altri è molto più facile. In pratica, quello che cambia fra un sistema e l'altro è quanti e quali metadati può gestire, quanto spazio e banda consuma per funzionare a parità di progetto e quali operazioni sono più rapide. I tre sistemi Scv Open Source per Linux presentati nei paragrafi che seguono danno un'idea adeguata dell'offerta in questo campo.

Bazaar (<http://bazaar.canonical.com>)

Si tratta di un Scv molto flessibile, che gira anche su OS X e Windows e viene proposto con lo stesso slogan di Ubuntu, "Controllo delle versioni per esseri umani", per sottolineare la facilità d'uso. Tecnicamente è di tipo distribuito, ma può essere configurato per lavorare in modalità centralizzata e funziona bene anche per singoli utenti. È anche utilizzabile come client per agganciarsi a progetti gestiti con altri sistemi, ad esempio Subversion.

Git (<http://git-scm.com>)

È stato scritto nientedimeno che da Linus Torvalds, l'autore originario e coordinatore del kernel Linux, in un momento di frustrazione causato proprio dai limiti degli altri Scv Open Source (secondo l'opinione di Torvalds). La sua architettura è prevedibilmente ottimizzata per progetti come Linux, distribuiti fra moltissimi programmatori quasi completamente indipendenti. Ogni nuova versione di un file, o dell'intero progetto, può essere descritta con etichette (tag) dettagliatissime, autenticabili con firma digitale. Git si può provare anche via browser all'indirizzo <http://try.github.io/levels/1/challenges/1>.

Subversion (<http://subversion.apache.org/>)

Questa soluzione è ottimizzata per gestire processi centralizzati. Per questo, anche quando la si usa su un solo computer, occorre creare una repository "ufficiale" in cui conservare i file. Varie funzioni consentono di raggruppare più repository preesistenti in una sola, o di suddividerne una se necessario. Anche in Subversion è possibile associare metadati complessi a ogni versione di ciascun file. Come Bazaar e Git, Subversion supporta i cosiddetti hook (alla lettera "uncini"): si tratta di script configurati in maniera tale da essere eseguiti automaticamente prima o dopo ogni operazione di estrazione o registrazione di file nel sistema.

Quando non serve un sito, ma un programma...

Gli ambienti di sviluppo descritti finora sono ottimi per creare siti che, pur non essendo gestibili con Cms standard, debbano essere fondamentalmente vetrine di contenuti. Come esempio di piattaforma Open Source per il secondo caso, ovvero la scrittura di vere e proprie applicazioni che elaborano dati in maniera più complessa, vogliamo invece citare Google Web Toolkit (Gwt, www.gwtproject.org).

GWT è l'insieme di codice, compilatori e altri strumenti software, tutto Open Source, alla base di applicazioni già diffusissime in tutto il mondo come i programmi AdSense e AdWords di Google, o della piattaforma Blogger. In generale Gwt consente di sviluppare e collaudare rapidamente servizi generici che girano via Internet su qualsiasi browser con interfacce e velocità di esecuzione paragonabili a quelle di programmi nativi per desktop.

Il programmatore Gwt scrive il suo codice in linguaggio Java ad alto livello, potendo utilizzare sia i componenti grafici (widget) del pacchetto sia quelli sviluppati in altri ambienti o generiche funzioni JavaScript. Quando il codice è pronto, il compilatore Gwt lo ottimizza in tutti i modi possibili, per poi ricavarne una serie di file JavaScript, ognuno fatto apposta per un diverso browser da desktop o mobile. Un singolo programma può anche essere automaticamente suddiviso in più frammenti, per consentire ai browser di partire con l'esecuzione non appena ha finito di scaricare il primo. Gwt è interessante per gli sviluppatori soprattutto per come facilita il *debugging* del codice. Questa operazione può infatti avvenire sia in un Ide tradizionale Java per desktop, sia in un qualsiasi editor di codice affiancato a un browser.

Il plugin chiamato Gwt Developer permette infatti ai browser di utilizzare direttamente il codice Java che si sta scrivendo, come se fosse il file JavaScript che sul campo riceverebbero da questo o quel server Web. A quel punto diventa anche possibile utilizzare tutti gli altri plugin per analisi e *debugging* di JavaScript già disponibili per Firefox, Chrome e altri browser. Chrome ne ha uno specifico per Gwt, con cui analizzare le prestazioni del codice così sviluppato e individuarne i punti critici.



«Lo sviluppo per il web richiede strumenti di collaudo specifici per verificare l'interazione con gli utenti.»

SeleniumHq, quando il browser collauda i siti per noi

Qualunque ambiente di progettazione e verifica si usi, nello sviluppo di applicazioni Web prima o poi si devono fare i conti con un ostacolo insormontabile: il software funziona davvero solo se e quando, girando nel browser dell'utente finale, reagisce correttamente alle azioni di quest'ultimo. Questo significa che non c'è scampo a prove manuali, noiosissime e interminabili, in cui si fare clic su tutti i link e pulsanti possibili per controllare cosa succederà? Fortunatamente no.

Il collaudo di alcune azioni o casi limite non può che essere manuale, ma la maggioranza di certe interazioni può essere verificata automaticamente

con strumenti Open Source come SeleniumHq (www.seleniumhq.org). La sua tecnologia, già integrata in molti browser, permette di controllare le loro azioni da codice scritto in molti altri linguaggi. In altre parole, SeleniumHq permette di scrivere programmi che "facciano clic" o scelgano in sequenza tutte le voci di un menu Html tramite browser, proprio come se un essere umano lo avesse lanciato e usato manualmente nello stesso modo nel suo desktop.

Sempre grazie a SeleniumHq quei programmi possono verificare cosa è accaduto nel browser in risposta a una certa azione e segnalare qualsiasi discordanza con i risultati attesi. In questo modo si possono sia verificare le interfacce Web di un'applicazione prima di rilasciarla, sia velocizzare

la soluzione di bachi segnalati dagli utenti: se una certa sequenza di clic manda in tilt un sito, la si può far ripetere a SeleniumHq su varie combinazioni di codice finché non si trova quella giusta.

Le due applicazioni, o meglio i modi più promettenti al momento per utilizzare Gwt potrebbero essere SmartClient (<https://smartclient.com>) e SmartGwt (<http://code.google.com/p/smartgwt/>). Il primo è una piattaforma che, in parole povere, permette di sviluppare sia il lato server sia quello client di una applicazione Web in sincronia. SmartClient genera automaticamente gran parte dell'interfaccia grafica (anche per terminali mobili!), a partire da dichiarazioni di quali strutture dati vanno usate e di come l'utente può leggerle o modificarle. SmartGwt, dal canto suo, è proprio quel che il suo nome suggerisce: una interfaccia di programmazione con cui usare le tecnologie Gwt e SmartClient in maniera integrata.

A metà fra Web e Desktop: Ubuntu Html5 Sdk

Concludiamo con una domanda difficile: dove comincia il Web e dove finisce il desktop? Devono proprio essere due ambienti diversi? Canonical, l'azienda che sponsorizza e dirige lo sviluppo della distribuzione Linux Ubuntu, già da diversi anni risponde con una singola interfaccia grafica (Unity) per qualsiasi dispositivo, dai desktop agli smartphone. Un risultato di questa strategia è un prodotto, sempre Open Source, che non possiamo non citare in questo articolo, anche se di sfuggita: un Software Development Kit (<http://developer.ubuntu.com/apps/html-5/>) creato specificamente per scrivere, con Html5, Css e JavaScript, programmi per desktop perfettamente integrati con, appunto, i desktop Ubuntu.

Le applicazioni sviluppate con Ubuntu Html5 Sdk girano in un cosiddetto *web container* che simula il browser e fornisce accesso diretto a tutto il resto dell'ambiente.

LINUX News

Verizon spinge il cloud Open Source

Verizon Terremark è un fornitore di servizi informatici di proprietà di uno dei maggiori fornitori di telefonia e accesso a Internet degli Stati Uniti. A marzo 2014 Terremark è entrata nella Linux Foundation come membro di categoria Oro, un livello che, per responsabilità e finanziamenti, include giganti come Google, China Mobile, Panasonic e Toyota, tanto per citare alcuni dei nomi più noti. La ragione di questo investimento è l'interesse di Terremark, e quindi di Verizon, nel cloud computing Open Source. Partecipando alla Linux Foundation Terremark intende innanzitutto promuovere l'uso di Verizon Cloud, una piattaforma per storage e applicazioni su cloud basata sull'ipervisor per Linux (gestore di macchine virtuali) chiamato Xen. Secondo i comunicati ufficiali, questo porterà anche a una accelerazione generale dello sviluppo di tecnologie Open Source per il cloud computing.

Australis, per un Firefox più su misura

Firefox 29, che dovrebbe essere disponibile a maggio 2014, sarà la versione di questo browser con i maggiori cambiamenti dell'interfaccia grafica da tre anni. Il pezzo forte del nuovo ambiente, chiamato Australis, dovrebbe essere un sistema per riarrangiare vari elementi dell'interfaccia stessa in maniera molto più flessibile che in passato. Anche le schede sono state ridisegnate, per identificarle più facilmente quando se ne aprono diverse decine. Dietro le quinte ci sarà anche una nuova versione di Firefox Sync, la funzione che permette di condividere e tenere sempre sincronizzati i propri bookmarks e altri parametri su diverse copie del browser. Firefox 29 dovrebbe infatti essere capace di gestire nuovi "Account Firefox", che consentiranno di effettuare quelle operazioni semplicemente inserendo email e password.