



Oliver Lau, Noud van Kruysbergen

Autonomie

Smartphones programmeren met HTML, CSS en JavaScript

Een universele app, die op Android, iPhone, BlackBerry, Symbian en Windows Mobile moet draaien, hoeft niet per se vijf keer opnieuw ontwikkeld te worden. Door gebruik te maken van de webtechnieken HTML, CSS en JavaScript werkt dezelfde code zonder aanpassingen op alle smartphones – in theorie in ieder geval.

Als niet de uitvoersnelheid maar de ontwikkelsnelheid de hoogste prioriteit heeft, dan zijn apps op basis van de webtechnieken HTML, CSS en JavaScript een goede optie. Voor het produceren van een universele app voor de verschillende platforms hoef je dan niet

bezig te houden met de verschillende SDK's en programmeertalen. Vereist zijn wat elementaire kennis van webtechnieken en een willekeurig JavaScript-framework. Dat laatste is nodig om de app de look & feel te geven die je van een 'normale' app gewend bent.

Een webapp wordt door de browser geladen. Je moet dus online zijn om hem te kunnen gebruiken. Een deel van de applicatielogica (voornamelijk het afhandelen van de gebruikersinterface) wordt in JavaScript gedaan. Als bijvoorbeeld formuliergegevens opgeslagen moeten worden, geeft de app die op de gebruikelijke wijze door aan een webserver.

Met technieken die te maken hebben met HTML5 kun je ook apps ontwikkelen die zonder webserver toch gegevens kunnen opslaan. Je kunt met JavaScript namelijk bij de globale variabele `window.localStorage` om sleutel/waardeparen te gebruiken en `openDatabase()` of `executeSql()` om een databasetabel aan te maken, te vullen en via SQL uit te lezen. Ook de applicatie zelf is via het attribuut `manifest` in de inleidende `<html>`-tag offline beschikbaar te maken. Meer over de vernieuwingen in HTML5 kun je vinden in [1].

Een brug slaan

Een webapp loopt dus in de browser en emuleert de look & feel van gewone apps zo goed en zo kwaad als dat kan met HTML, CSS en wellicht wat afbeeldingen. Toegang tot sensoren, netwerkinterfaces en audio/video is daarmee helaas niet mogelijk, omdat JavaScript geen interfaces voor deze besturingssysteemfuncties heeft. Om ervoor te zorgen dat dat wel werkt, moet er nog een bemiddelaar zijn die de functies voor JavaScript ontsluit.

Dat is precies wat er gebeurt bij de quasi-originele apps. Die gebruiken eveneens webtechnieken, maar bundelen alle onderdelen tot een afgesloten app, die in de webengine van het betreffende platform wordt uitgevoerd. De webengine fungeert dan ook als bemiddelaar tussen JavaScript en het besturingssysteem. Als een quasi-originele app aan bepaalde voor-

waarden voldoet, wordt het ook wel een widget genoemd [2].

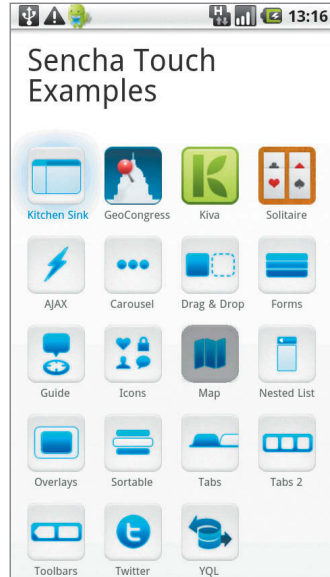
Deze quasi-originele apps hebben meerdere voordelen. Je hoeft niet meer online te zijn om de app te kunnen gebruiken, en je kunt die app als ontwikkelaar net als iedere andere app als eenheid via de bekende distributiekanaalen aan de man brengen.

In dit artikel gaan we in op de wat ons betreft aantrekkelijkste en populairste frameworks en ontwikkelomgevingen voor het programmeren van JavaScript-apps. Voor geen enkele van deze producten hoef je te betalen voor privé-gebruik. Alle genoemde software en links naar tutorials zijn te vinden via de softlink aan het eind van dit artikel.

Ad hoc

Maar eerst nog even wat over app-generators waarmee je met een paar muisklikken een kant-en-klare app kunt toveren. Dat klinkt natuurlijk allemaal geweldig, maar je oorspronkelijke euforie wordt waarschijnlijk al snel in de kiem gesmoord door de zeer beperkte mogelijkheden. Er zijn generators voor RSS/Atom-feeds en een paar voor een mediagalerij. En dat was het dan.

Nokia biedt voor Symbian-toestellen de Ovi App Wizard aan. Na registratie hoef je alleen maximaal vier feed-urls en een naam voor de app in te voeren en een pictogram te uploaden. Bovendien kun je een kleur kiezen voor de achtergrond, tekst en links. Als je reclame in je app wilt hebben,



Sencha Touch biedt een aantal JavaScript-frameworks en kant-en-klare pictogrammen om een webapp er in ieder geval uit te laten zien als een originele app.

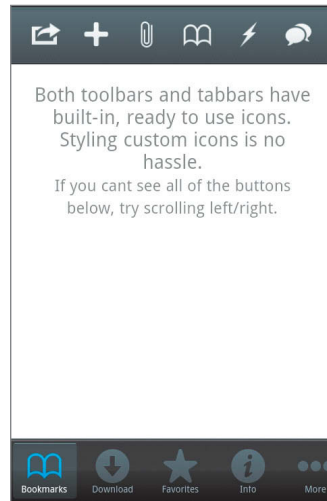
kun je kiezen uit de diensten Millennialmedia en Mpression. Vervolgens wordt de app in de Ovi Store gezet en kan die daar na goedkeuring van Nokia na korte tijd gedownload worden.

Opera bewandelt een vergelijkbare weg met z'n widget-generator Widgetize. Met skins van hoogglans zwart, roze pluche en tijgervel tot geborsteld metaal kun je de RSS-Operawidget een eigen tintje geven. De mediagalerij-widget laat de foto's van een Flickr-account zien of video's van een YouTube-kanaal.

Tot voor kort was er ook voor de iPhone een app-generator, maar producent AppLoop heeft deze aanvankelijk zeer geprezen dienst intussen beëindigd. Het is sowieso maar de vraag hoe lang Apple automatisch gegenereerde apps nog in de App Store duldt en niet weert vanwege trivialiteit.



Met een RSS-widget van Opera kun je het nieuws van de harde werkelijkheid een zacht randje geven.



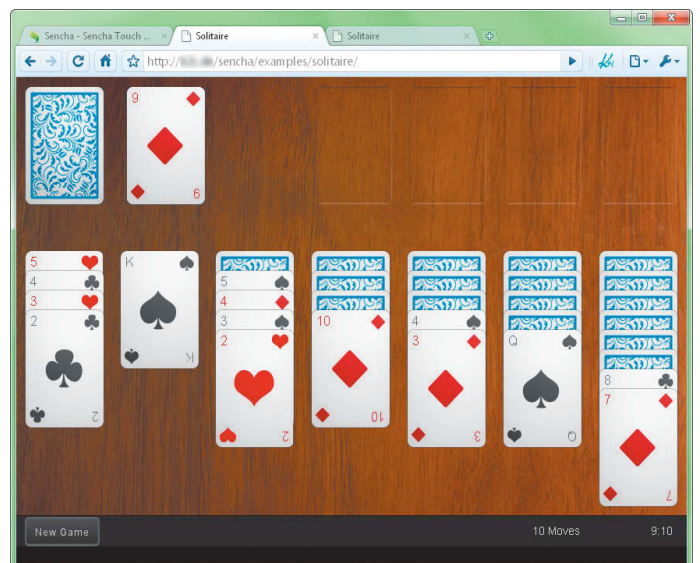
Sencha Touch

Als je iets wilt hebben dat echt van jezelf is, zul je het zelf moeten programmeren. Het hage-nieuwe framework Sencha Touch verenigt de JavaScript-frameworks Ext JS, Raphaël en JQTouch. Het claimt het eerste framework te zijn dat features uit HTML5 en CSS3 gebruikt. HTML5 is bijvoorbeeld handig voor het afspelen van audio- en videobestanden, CSS3 voor het tekenen van afgeronde hoeken, schaduw en kleurverlopen.

Het JavaScript-framework Ext JS stelt browseronafhankelijke functies ter beschikking voor DOM-traversal (Document

Object Model), het afhandelen van events en asynchrone communicatie (Ajax). Raphaël dient voor het weergeven van vectorafbeeldingen. In een WebKit-browser gebeurt dat met SVG, bij Internet Explorer met VML. JQTouch is een uitbreiding van de beroemde JavaScript-bibliotheek JQuery [3] voor touch-interfaces.

Een aantal voorbeelden laten zien hoe krachtig Sencha Touch is. Je kunt gegevens via JSONP, YQL (Yahoo Query Language) en Ajax laden, Google-Maps-kaarten weergeven en de layout met overlays, werkbalken, sorteerbare en geneste lijsten, tabbladen, scrollen (ook kine-



Het ziet eruit als een gewone desktopapplicatie, maar in werkelijkheid is het een app op basis van Sencha Touch. Voor de huidige mobiele displays is de app echter te breed en te hoog, waardoor hij alleen goed te gebruiken is op de iPad.

```
Ext.setup({
  tabletStartupScreen:
    'tablet_startup.png',
  phoneStartupScreen:
    'phone_startup.png',
  icon: 'icon.png',
  glossOnIcon: false,
  onReady: function() {
    // Create a Carousel of Items
    var carousel1 =
      new Ext.Carousel({
        defaults: {
          cls: 'card'
        },
        items: [{
          html:
            '<h1>Carousel</h1>'
        },
        {
          title: 'Tab 2',
          html: '2'
        },
        {
          title: 'Tab 3',
          html: '3'
        }
      ]
    );
    var carousel2 =
      new Ext.Carousel({
        direction: 'vertical',
        ui: 'light',
        defaults: {
          cls: 'card'
        },
        items: [{
          title: 'Tab 1',
          html:
            '<h1>ui= light </h1>'
        },
        {
          title: 'Tab 2',
          html: '2'
        },
        {
          title: 'Tab 3',
          html: '3'
        }
      ]
    );
    new Ext.Panel({
      fullscreen: true,
      layout: {
        type: 'vbox',
        align: 'stretch'
      },
      defaults: {
        flex: 1
      },
      items: [carousel1,
        carousel2]
    });
  });
});
```



Een beetje HTML, wat CSS en een paar regels JavaScript – meer heb je niet nodig om met Sencha Touch een grafische bedieningsinterface te maken waarmee je met vingerbewegingen nieuwe elementen (zogenoemde cards) heen en weer kunt schuiven.

tisch) en overgangseffecten voor pagina's verfraaien. Fraaie pictogrammen voor bijna ieder doel kunnen je eigen webapp optisch nog meer cachet geven.

Het enorme aantal functies heeft ook een nadeel: ondanks compressie komen de JavaScript-bestanden met een grootte van meerdere honderden kilobytes op de webserver te staan en dus zal het even duren voordat die binnen zijn bij een mobiel apparaat, wat bij een langzame GPRS/EDGE-verbinding te merken is aan onaangename wachttijden.

Helaas werkt de bediening alleen op een iPhone/iPad echt vloeierend en zonder haperingen. Bij sommige WebKit-compatibele mobiele browsers is de weergave van CSS3-lay-outs schokkerig, bijvoorbeeld bij randen van een afbeelding (border-image). Sommige Android-browsers (ook bij een Android 2.2-telefoon als de Google Nexus One) en verrassend genoeg ook Safari op de iPhone vervangen dergelijke afbeeldingen soms door een eenkleurige achtergrond.

PhoneGap

PhoneGap is een platformafhankelijk framework dat functies van het besturingssysteem aan JavaScript beschikbaar stelt. Daardoor ontstaan quasi-originele apps met precies dezelfde look & feel als je van het betreffende platform gewend bent. Je kunt bijvoorbeeld het mobieltje laten trillen of piepen, audio of video afspelen, de locatie achterhalen, de versnellingssensor uitlezen, in de contactdatabase kijken of foto's maken met de camera. Bovendien heeft de API de JavaScript-bibliotheek XUI, die speciaal voor mobiele applicaties tot het noodzakelijke gestript is. Daar kun je bijvoorbeeld de DOM mee doorzoeken en modifieren, events zoals vingerbewegingen mee verwerken of een link laden via XMLHttpRequest.

Gek genoeg zijn niet alle API-functies voor alle platforms beschikbaar. Een piepje geven kan wel met iPhone, Android en BlackBerry, maar niet met Symbian. Je kunt de huidige locatie wel op alle platforms uitlezen, maar alleen onder Symbian kun je bij iedere positieverandering automatisch een callback-functie

laten oproepen. Deze onbalans zit in de hele API, waardoor een app die met PhoneGap is gemaakt volgens Murphy's Law wellicht op maar één platform werkt. Gelukkig heeft PhoneGap niets tegen andere bibliotheken, waardoor je bijvoorbeeld ook Sencha Touch kunt gebruiken, dat wat meer uitgerijpt is op JavaScript-gebied.

PhoneGap heeft tools om een appfundering (een platformspecifiek app-omhulsel met HTML/CSS/JavaScript-basis) in een map naar keuze aan te maken. Wat je als ontwikkelaar met die fundering doet, moet je zelf weten. iPhone-ontwikkelaars starten er waarschijnlijk een Xcode-project mee, voor Android en BlackBerry zul je Eclipse gebruiken. Als het doelplatform Symbian is, kun je het project in de WRT IDE importeren. Daarover zodadelijk meer. Voor Windows Mobile is er nog geen gedocumenteerde oplossing.

De bij het framework horende simulator is erg gaaf. Je kunt kiezen uit tien goed nagemaakte smartphones. Op de achtergrond werkt als browser Firefox, die met een geactiveerde Firebug-add-on meteen ook dienst doet als debugger.

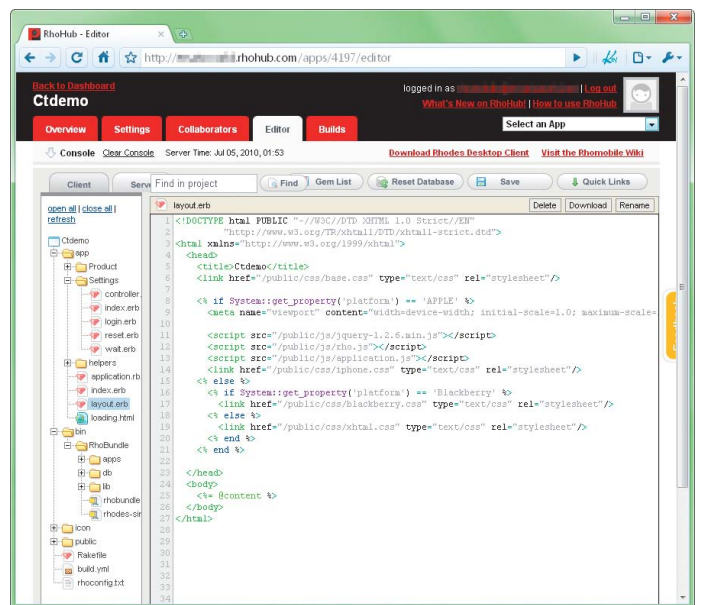
De documentatie valt echter tegen. In de Spartaanse wiki staan wat tips en trucs, die grotendeels verouderd zijn. De API-referentie is verdeeld over een handjevol webpagina's, waarin

meer witruimte staat dan direct bruikbare informatie. Soms tref je een stukje code aan dat als voorbeeld dient voor hoe je een functie moet gebruiken, dan weer alleen een opsomming van de parameters. Soms is het zelfs voor JavaScript-cracks niet duidelijk hoe functies gebruikt moeten worden.

Rhodes

Ook Rhodes hoort tot de frameworks voor quasi-originele apps. Je kunt er mee ontwikkelen voor iPhone/iPad, Android, BlackBerry, Symbian en Windows Mobile. Het is echter uitzonderlijk dat de applicatielogica niet in JavaScript maar in Ruby geprogrammeerd moet worden. Net als bij ASP.NET, PHP en embedded Perl wordt de Ruby-code in HTML ingebed.

Maar eerst moet je door een heel installatiecircuit heen: Ruby, Ruby Gems, C/C++-compiler (bij Windows is dan ook GnuWin32 nodig) en dan tot slot nog Rhodes. Ook bij het aanmaken van een projectbasis voor een bepaald platform moet je wat geduld hebben omdat het set-upprogramma de compiler opstart om de runtime-omgeving voor de app te compileren. Je kunt een app met een editor naar keuze maken. Het handigst is Eclipse met de Ruby-plug-in. Je kunt de app testen in de emulator van het gewenste platform.



Bij rhohub.com kun je Rhodes-projecten maken, beheren, compileren en bewerken. De online editor kleurt de sleutelwoorden van de in HTML ingebodde Ruby-code in.



In de PhoneGap-simulator kun je apps die op PhoneGap gebaseerd zijn op verschillende platformen proberen en debuggen.

en de Dopod A6188, een voor de Chinese markt aangepaste HTC Magic met Open Mobile System in plaats van Android. De joint-venture-ondernemers hebben echter ook wat andere modellen aangekondigd die JIL-widgets moeten kunnen gaan uitvoeren.

De runtime-omgeving kun je wel op Android- en Symbian-telefoons (v3 en v5) zetten. Die zit in de Vodafone App Manager, die je ook voor het installeren van een widget nodig hebt. Dat is echter redelijk omslachtig als de widget bijvoorbeeld op je pc thuis staat en op een Android-telefoon gezet moet worden. Dan moet je de widget eerst naar de geheugenkaart van het mobieltje kopiëren en

Je kunt een nieuwe app ook met het online ontwikkelplatform RhoHub maken, bewerken, compileren en inpakken. Daarbij is 50 MB online opslagruimte gratis. Die zijn echter vrij snel vol, omdat alleen de binary van een elementaire hello-world-app door de runtime-omgeving van Ruby al 1,6 MB groot is – als er überhaupt een binary ontstaat: het door ons gestarte build-proces was na twee dagen nog steeds bezig...

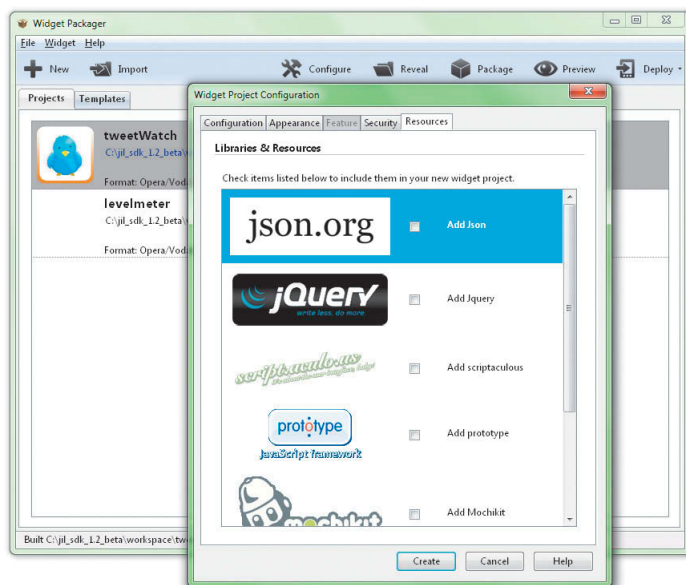
JIL

De JIL SDK van het Rotterdamse Joint Innovation Lab (een joint-venture van Vodafone, Verizon Wireless, China Mobile en Softbank Mobile) fungeert als een ontwikkelomgeving voor

widgets. Het is gebaseerd op de niet geheel meer okselrijse versie 3.5.1 van de open-source ontwikkelomgeving Eclipse. Behalve een JavaScript-bibliotheek en wizards om snel een project te kunnen starten, krijg je ook een Android-emulator. De API laat amper wat te wensen over: toegang tot de camera, gps, versnellings-sensor, mediaspeler, agenda, adresboek en bestandstelsysteem, alles zit erbij en is gedocumenteerd. Al is dat laatste soms wel beperkt en soms zonder kennis van het Japans moeilijk te begrijpen.

JIL-widgets zijn niets anders dan een Zip-bestand met als extensie .wgt, waarin de HTML/CSS/JavaScript-code zit, samen met de gebruikte afbeeldingen

en een XML-bestand met wat meta-informatie, zoals de naam van de app, het versienummer en de auteur. Om ervoor te zorgen dat deze widgets op een telefoon werken, heb je een speciale runtime-omgeving nodig. Die runtime-omgeving is er op dit moment echter maar voor twee telefoons: de Samsung H1



Als je Eclipse te onhandig vindt als ontwikkelomgeving, hoef je toch niet zonder projectwizard te zitten: de Vodafone Widget Packager helpt bij het maken en configureren van een nieuw widget-project en het inpakken en publiceren ervan.

```
<html>
<head>
  <meta name= viewport content= width=320; user-scalable=no />
  <meta http-equiv= Content-type content= text/html; charset=utf-8 >
  <title>PhoneGap</title>
  <link rel= stylesheet href= master.css type= text/css
media= screen
  title= no title charset= utf-8 >
  <script type= text/javascript charset= utf-8
src= phonegap.js ></script>
  <script type= text/javascript charset= utf-8 >
var deviceInfo = function() {
  document.getElementById( platform ).innerHTML = device.platform;
  document.getElementById( version ).innerHTML = device.version;
  document.getElementById( uuid ).innerHTML = device.uuid;
  console.log( Height: + window.innerHeight);
  console.log( Width: + window.innerWidth);
}

var getLocation = function() {
  var suc = function(p){
    alert(p.coords.latitude + + p.coords.longitude);
  };
  var fail = function(){};
  navigator.geolocation.getCurrentPosition(suc,fail);
}

var beep = function(){
  navigator.notification.beep(2);
}

var vibrate = function(){
  navigator.notification.vibrate(0);
}

var getContact = function(){
  var suc = function(c){ alert( Contact 4: + c.contacts[3].name); };
  var fail = function(){};
  navigator.ContactManager.get(suc, fail);
}

var watchAccel = function() {
  var suc = function(a){
    document.getElementById('x').innerHTML = roundNumber(a.x);
    document.getElementById('y').innerHTML = roundNumber(a.y);
    document.getElementById('z').innerHTML = roundNumber(a.z);
  };
  var fail = function(){};
  var opt = {};
  opt.frequency = 100;
  timer = navigator.accelerometer.watchAcceleration(suc,fail,opt);
}
// ...
```

PhoneGap voegt de besturingssystemfuncties binnen JavaScript toe als objecten in de object navigator.

```
Titanium.UI.setBackgroundColor('#000');
var tabGroup = Titanium.UI.createTabGroup();
var win1 = Titanium.UI.createWindow({
  title:'Tab 1',
  backgroundColor:'#fff'
});
var tab1 = Titanium.UI.createTab({
  icon:'KS_nav_views.png',
  title:'Tab 1',
  window:win1
});
var label1 = Titanium.UI.createLabel({
  color:'#999',
  text:'I am Window 1',
  font:{fontSize:20,fontFamily:'Helvetica Neue'},
  textAlign:'center',
  width:'auto'
});
win1.add(label1);
var win2 = Titanium.UI.createWindow({
  title:'Tab 2',
  backgroundColor:'#fff'
});
var tab2 = Titanium.UI.createTab({
  icon:'KS_nav_ui.png',
  title:'Tab 2',
  window:win2
});
var label2 = Titanium.UI.createLabel({
  color:'#999',
  text:'I am Window 2',
  font:{fontSize:20,fontFamily:'Helvetica Neue'},
  textAlign:'center',
  width:'auto'
});
win2.add(label2);
tabGroup.addTab(tab1);
tabGroup.addTab(tab2);
tabGroup.open();
```

Het Titanium-framework heeft een aantal functies voor het maken van GUI's, waaronder tabbladen.

dan met de filemanager van de telefoon (bijvoorbeeld Astro File Manager) naar de betreffende locatie gaan om het bestand aan te klikken. Daarna opent de Vodafone App Manager en wordt de widget geïnstalleerd.

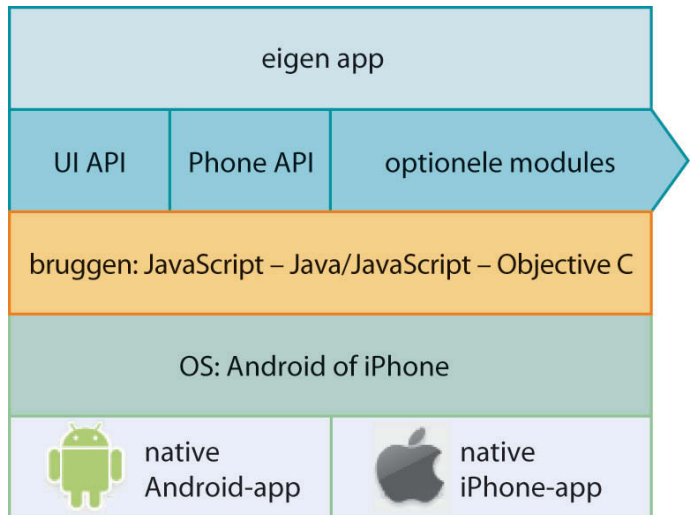
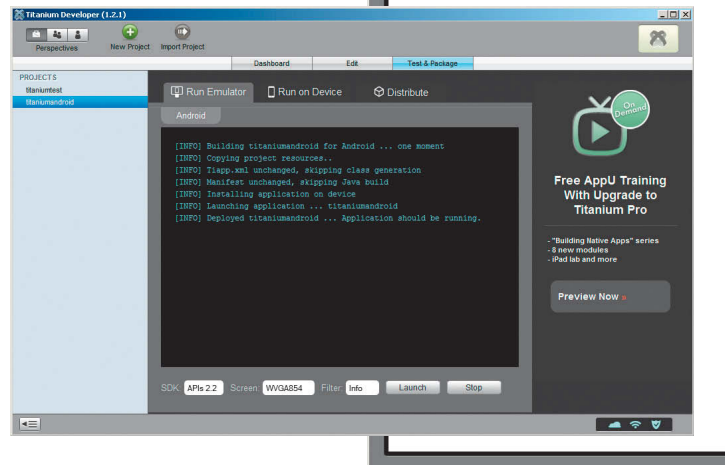
So far, so good. Laten we hopen dat JIL de documentatie nog wat meer uitbreidt, de runtime-omgeving voor andere platforms gaat aanbieden en het installeren van widgets op het mobieltje makkelijker wordt.

Appcelerator

Het Softwarebedrijf Appcelerator uit Californië ziet zijn gratis verkrijgbare Titanium als een directe concurrent voor Adobe AIR. Als je naar de doelgroep kijkt, gaat Titanium zelfs nog verder: AIR ondersteunt alleen het ontwikkelen van desktopapplicaties, Titanium Developer richt zich ook op mobiele applicaties. De desktopapplicaties van Titanium werken onder Windows (32-bit), Mac OS X (Intel) en Linux. De mobiele applicaties onder iPhone OS 3.1 en hoger (inclusief iPad) en Android 1,6 en hoger. Ondersteuning voor BlackBerry zit als bètaversie in de commerciële variant Titanium Pro.

Het HTML5- en CSS3-framework kun je bij de desktopapps vullen met applicatielogica op basis van JavaScript, Python of Ruby. Mobiele apps zijn alleen met JavaScript mogelijk. Titanium bestaat uit twee delen. Ten eerste een JavaScript-framework met programmeerinterfaces voor het maken van gebruikersinterfaces, voor het bepalen van de locatie, voor

Appcelerator Titanium ondersteunt de ontwikkelaar bij het maken, inpakken en publiceren van een app. Er zit geen JavaScript/HTML/CSS-editor bij met syntax highlighting en dergelijke dingen.



Een app die je met webtechnieken ontwikkeld hebt, heeft middleware als bijvoorbeeld Titanium nodig om een brug te slaan tussen de JavaScript-code en het besturingssysteem.

het uitlezen van de versnellingsensor, toegang tot het bestandssysteem, het herkennen van gestures, het tekenen van kaarten, voor sociale netwerken met Facebook en Yahoo en nog veel meer. Het tweede deel is een GUI-applicatie met een projectwizard, een webengine voor het testen van desktopapplicaties en tools om van de broncode een app te maken – maar helaas geen editor. Appcelerator vindt dat zelf geen tekortkoming maar een feature omdat je dan je lievelingseditor kunt gebruiken. Als je die nog niet hebt, kun je eens naar de ontwikkelomgeving Eclipse kijken.

Appcelerators advertentieslogan is "Our apps are native" – en dat klopt ook. JavaScript-frameworks als JQTouch en dergelijke werken met HTML en CSS en afbeeldingen om de look & feel van Android, iPhone en dergelijke na te maken, maar bij Titanium ontstaan GUI-elementen door het aanroepen van een besturingssysteemfunctie. Appcelerator heeft daar wrapper-functies voor ingebouwd die het besturingssysteem toegankelijk maken voor JavaScript. Als je bijvoorbeeld een object Titanium.UI.ProgressBar maakt, dan verschijnt op het beeldscherm precies zo'n voortgangsbalk als je van andere An-

droid- of iPhone-apps kent en geen zo goed of zo kwaad als mogelijk nagemaakte variant.

Als Titanium-gebruiker krijg je toegang tot een webpagina waar Appcelerator op zet hoeveel gebruikers in de afgelopen zeven dagen de app gedownload hebben. Als je dat over een langere termijn wilt kunnen bekijken, moet je een betaalde licentie nemen, maar dat is met prijzen vanaf 200 dollar per maand niet echt goedkoop. In die licentiekosten zit een 'Premium'-support via de website van Appcelerator.

Met behulp van de Titanium-API kun je bovendien het gedrag van de gebruikers registreren en voor statistische analyse naar de website sturen. Daardoor krijg je inzicht in welke leeftijdsgroepen je app gebruiken, waar ze vandaan komen, hoe lang ze die app gebruiken en welke functies, wanneer ze waar vandaan op Facebook inloggen en welke muziek en video ze bekeken hebben. Als je enigszins gevoelig bent voor privacy-issues, gaan je haren daar wel recht van overeind staan, maar Appcelerator heeft geen enkele scrupules om gebruikers zo te bespioneren. Nergens op de website van Appcelerator, in de API-documentatie of op de supportportal is er iets te vinden over hoe ontwikkelaars met de privé-sfeer van gebruikers om moeten gaan. Appcelerator belooft alleen de verstuurde gegevens zorgvuldig te behandelen.

Symbian Web Development Tools

Symbian OS heeft sinds versie 5 een runtime-omgeving voor webapplicaties. Nokia's Web Runtime 1.0, zoals de officiële naam luidt, gebruikt in principe de HTML/CSS/JavaScript-engine WebKit, waarbij als standaarden gewerkt wordt met HTML 4.01, XHTML mobile profile, CSS 2.1 en JavaScript 1.5. Om toegang te krijgen tot besturingssysteemcomponenten als gps, audio en video is er de JavaScript-interface PhoneGap aan toegevoegd. De HTML-, CSS- en JavaScript-code is samen met de Web Runtime tot een kant-en-klare Symbian-app te smeden, die aan de buitenkant niet van een originele app te onderscheiden is.

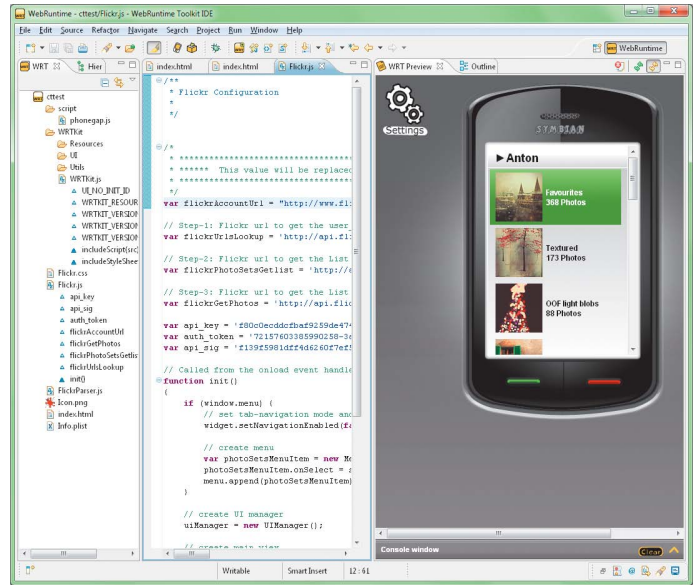
Dat doe je met de Symbian Web- Runtime Tools (WRT Tools). Deze worden in twee vormen aangeboden: als een plug-in gebaseerd op de Eclipse-distributie Aptana en als een afzonderlijke Eclipse-installatie (Nokia WRT IDE). Helaas zit de Symbian-simulator alleen in de WRT IDE. Veranderingen aan de broncode van de app kun je daar meteen mee testen.

Door de ingebouwde projectwizard zijn de eerste ontwikkelstappen makkelijk: je kunt kiezen uit een eenvoudige basis (met het JavaScript-framework WRTKit) en twee voorbeeldprojecten: een Flickr-fotoalbum en een RSS-feedreader. WRTKit biedt functies voor het programmeren van de bedieningsinterface. De GUI's die je met WRTKit maakt, lijken erg goed op wat bij Symbian gebruikelijk is.

Ook de documentatie is het noemen waard. Er is niet alleen een uitvoerige beschrijving van de API, maar er zijn ook tutorials om snel aan de slag te gaan en artikelen die de begrippen uitleggen. Al wordt die info niet handig centraal aangeboden. Soms moet je kijken op developer.symbian.org, dan weer op www.forum.nokia.com. Met een paar bookmarks in je browser is dat echter geen onoverkomelijk probleem.

Gemengde gevoelens

Als je apps voor meerdere platforms wilt ontwikkelen, hoeft je door het gebruik van bekende webtechnieken niet ook nog eens met tig programmeertalen en frameworks te leren werken. Kennis van HTML, CSS en JavaScript is voldoende.



De WebRunTime Toolkit IDE is gebaseerd op Eclipse en heeft een simulator voor Symbian geïntegreerd in de HTML/CSS/JavaScript-ontwikkelomgeving.

JIL is helaas nog niet volwassen genoeg om serieus te overwegen. De Symbian-oplossing van Web Development Tools en WRT-ontwikkelomgeving is fraai en krachtig, maar niet platformafhankelijk. Rhodes is een potente ontwikkeltool voor bijna alle platforms. Alleen is het een vreemde eend in de bijt omdat er met Ruby gewerkt wordt. Als je alleen van plan bent voor de meest populaire platforms Android en iPhone te ontwikkelen, zou je vanwege de geslaagde manier van inpakken en de fraaie interfaces eens naar Titanium Developer moeten kijken.

De enige twee frameworks die met alle platforms overweg kunnen, zijn Sencha Touch en PhoneGap. PhoneGap heeft

het voordeel dat het quasi-originele apps kan produceren, terwijl het met Sencha Touch makkelijker is om mooie GUI's te maken. Maar gelukkig kun je ze met elkaar combineren. (nkr)

Literatuur

- [1] Herbert Braun, Nieuw terrein, Wat brengt HTML5 voor webontwikkelaars, c't 3/2010, p.102
- [2] The Widget Interface, www.w3.org/TR/widgets-apis
- [3] Herbert Braun, Compacte scripts, JavaScript-bibliotheek jQuery in de praktijk, c't 10/2009, p.112
- [4] Jonathan Stark, Building iPhone Apps with HTML, CSS, and JavaScript: <http://building-iphone-apps.labs.oreilly.com>



JavaScript-Frameworks voor mobiele apps

Naam	Sencha Touch	PhoneGap	Rhodes	JIL	Titanium Developer	WRT
App-type	Web-App	quasi-native	quasi-native	quasi-native	quasi-native	quasi-native
Producent	Sencha	Nitobi	Rhomobile	JIL	Appcelerator	Symbian.org
Besturingssysteem	allemaal (Eclipse)	allemaal (Eclipse)	Windows, Mac OS X	allemaal (Eclipse)	Windows, Linux, Mac OS X	allemaal (Eclipse)
Licentie	GPLv3	MIT	MIT	JIL Widget SDK License (opensource)	Apache	Eclipse Public License
Programmeertaal (HTML, CSS ...)	JavaScript	JavaScript	Ruby	JavaScript	JavaScript	JavaScript
Platforms						
Android	✓	✓	✓	✓	✓	-
iPhone	✓	✓	✓	-	✓	-
WebOS	✓	✓	-	-	-	-
Symbian	✓	✓	✓	✓	-	✓
Blackberry	✓	✓	✓	-	✓ (alleen Pro-versie)	-
Windows Mobile	✓	✓	✓	-	-	-
✓ aanwezig	- niet aanwezig					