



door Georges Tarbouriech
<georges.t(at)linuxfocus.org>

Gorm en ProjectCenter, de RAD tools van GNUstep



Over de auteur:

Georges is reeds lang Unix gebruiker. Hij houdt van GNUstep en de tools die dit fantastische framework levert.

Kort:

RAD staat voor Rapid Application Development (snel applicaties ontwikkelen). Aan het eind van de jaren '80 toen NeXTstep werd gelanceerd, kwam het met een ongelofelijke tool, genaamd InterfaceBuilder. In combinatie met een andere tool, genaamd ProjectBuilder, was het mogelijk om applicaties in een oogwenk te bouwen. GNUstep biedt een vrije versie van deze tools, genaamd Gorm.app en ProjectCenter.app.

Vertaald naar het Nederlands door:

Guus Sniijders
<ghs(at)linuxfocus.org>

In den beginne...

Sinds de prehistory van computers is het ontwikkelen van software een enorme uitdaging geweest. Computers waren omvangrijk, maar met weinig vermogen. Ze waren nogal duur, er waren er niet echt veel en ontwikkelaars waren niet in staat om ze vaak te gebruiken, daar ze ze moesten delen met anderen. Toen probeerden onderzoekers om een manier te vinden om computers meer dan een taak tegelijk uit te voeren, om zo de efficiëntie te verhogen. Uiteraard moesten ze programmeer talen ontwerpen en bouwen, en daarin rekening houden met de beperkte resources van de beschikbare computers.

Zodoende verschenen in de jaren 60 verschillende nieuwe talen: LISP, FORTRAN, BASIC, Algol68, BCPL, enz. Daarna kwam de taal B, ontwikkeld vanuit de genoemde BCPL, en werd al snel de C taal. Dit laatste veranderde de wereld van programmeren.

De Object georiënteerde talen (SmallTalk, Objective C, C++, enz) verschenen later, tijdens het "grafische era".

In de jaren 80 boden sommige machines graphische besturingssystemen (Apple Macintosh, Commodore, Amiga, Atari ST, enz) en het X Window System werd ontwikkeld. Rond dezelfde tijd was een bedrijf bezig met een GUI voor IBM OS2, genaamd Presentation Manager. Voordat dat klaar was, lanceerde dit bedrijf een "eigen" GUI voor zijn DOS, genaamd... Windos. De eerste twee versies waren

nauwelijks bruikbaar, maar... de derde begon alles. De MvAI (Microsoft very Artificial Intelligence (Microsoft erg Kunstmatige Intelligentie)) was geboren!

Dat is, iedere gebruiker werd een computer wetenschapper, sindsdien hebben we "grote" applicaties gezien, geschreven met Excel of Word en Visual Basic :-)

Geen punt!

Lang voordat deze situatie ons bereikte, was NeXTstep gelukkig geboren, en daarmee kwam Interface Builder. Deze tool staat je toe om een GUI te bouwen voor je applicatie in een erg korte tijd en met erg veel gemak. Vanaf daar heeft dit soort tools zich verspreid. Tools zoals Omnis, 4D, Delphi, Kylix, enz, om er een paar te noemen. Sommige hiervan zijn multiplatform terwijl de meerderheid toegewijd is aan Windos. Merk op dat er ook vrije toolkits zijn met een dergelijke filosofie, Gtk (Gimp Tool Kit) bijvoorbeeld. Commerciële Unixen bieden ook dit soort tools.

De belangrijkste feature van deze tools is dat je de code voor de 200 vensters van je applicatie niet hoeft te schrijven, maar alleen degene die de data beheerd.

Of dit soort tools je aanstaan of niet is niet het punt. De ontwikkel tijd is kort: het is een feit (vandaar de naam, "Rapid Application Development").

GNUstep voorziet ons van vrije RAD tools. Ze heten Gorm en ProjectCenter. Natuurlijk, deze tools zijn erg "jong", maar ze werken. Laten we eens naar ze kijken.

GNUstep

Om Gorm en ProjectCenter te kunnen gebruiken, dien je ook GNUstep te installeren. Dit valt echter buiten dit artikel. Alles wat je nodig hebt is te vinden op de GNUstep website. Hier zijn broncode, HOWTOs, tutorials, enz te vinden.

Je kunt ook deze artikelen lezen: GNUstep, de open source OpenStep en GNUMail.app, het bewijs van portabiliteit.

De tests voor dit artikel zijn gedaan onder FreeBSD 4.7 met Window Maker 0.80.1, met gnustep-make-1.5.0, gnustep-base-1.5.0, gnustep-gui-0.8.2 en gnustep-back-0.8.2. Deze laatste zijn de nieuwste GNUstep unstable versies. Als je wilt, kun je ook de stable versies gebruiken. Ten slotte hebben we gcc 3.0.4 als compiler gebruikt.

Gorm.app

Gorm staat voor Graphic Object Relationship Modeler (of wellicht GNUstep Object Relation Modeler, zoals staat in het README bestand). Het is kloon van de eerder genoemde NeXTstep Interface Builder (of vandaag de dag MacOS X).

Gorm is het werk van Richard Frith-Macdonald die het project startte. Gregory Casamento is de huidige ontwikkelaar en doet het meeste van het werk met Pierre-Yves Rivaille. De huidige versie is 0.1.9.

Nieuwere CVS snapshots zijn beschikbaar vanaf <http://savannah.gnu.org/projects/gnustep>.

De nieuwste stabiele versie is te downloaden vanaf de GNUstep website.

De filosofie achter Gorm (en Interface Builder) is om de gebruiker te voorzien van objecten, gevonden in pallets, en deze objecten naar lege vensters te slepen om de grafische componenten van je applicatie te ontwerpen.

Deze objecten kunnen knoppen, velden, aanvink-vakjes, panelen, enz. Je kunt alles aan venster

toevoegen om deze gebruikersvriendelijker te maken. Vervolgens kun je deze aanpassen met inspectors. Vanuit de inspectors kun je attributen veranderen, connecties definiëren, grootte, help en classes voor de geselecteerde objecten aanpassen. Na het aanmaken van een class kun je outlets en acties toevoegen aan de objecten.

Vervolgens instantieer je de class, wat een nieuw object creëert (de instantie) in het main venster van Gorm, en verbindt je de outlets en de acties aan de corresponderende componenten. Dit kun je doen door met de muis van de instantie naar het geselecteerde object te slepen om outlets te verbinden en van het object naar de instantie om acties te verbinden. Ten slotte creëer je het geraamte van de class bron bestanden, en je bent klaar. Later meer hierover.

ProjectCenter.app

ProjectCenter is het "hart" van een project. Het is een kloon van Project Builder uit NeXTstep en Mac OS X.

ProjectCenter is het werk van Philippe C.D.Robert en de huidige versie is 0.3.0. Net als Gorm, kun je deze downloaden van de GNUstep website, bij de Developer apps sectie. Uiteraard kun je ook het laatste CVS snapshot gebruiken: dat hebben we voor dit artikel ook gedaan, dit was versie 0.3.1.

Vanuit ProjectCenter kun je een project creëren, de interface (met Gorm), de broncode schrijven; je kunt dit project compileren en uitvoeren (debug is nog niet beschikbaar). In het kort kun je alle resources die voor het project nodig zijn beheren: broncode, documentatie, bibliotheken, subprojecten, interfaces, enz.

Als je een nieuw project creëert, kun je het type aangeven. Je kunt kiezen uit applicatie, bundel, tool, bibliotheek en Gorm applicatie.

Naast andere dingen levert ProjectCenter een editor waarin je de Gorm skelet code kunt completeren.

Hoe werken Gorm en ProjectCenter samen? Erg goed, dank je!

Serieuzer, we zullen twee voorbeelden gebruiken om dit te illustreren.

Een paar voetnoten

Dit artikel is GEEN tutorial. Het idee is om te laten zien hoe gemakkelijk het gebruik van deze tools is, en tegelijkertijd om te demonstreren hoe je dezelfde code kunt gebruiken voor GNUstep (dus vele Unix platformen... en, als je van "worstelen" houdt, ook voor Windos) en MacOS X. Het enige dat je hoeft te doen is de interface op ieder platform ontwerpen daar de nib (InterfaceBuilder of Gorm) bestanden niet overdraagbaar zijn (tenminste, nog niet).

Het eerdergenoemde artikel over GNUMail.app liet de overdraagbaarheid vanuit het oogpunt van de gebruiker zien. Dit artikel richt zich vooral op het oogpunt van de ontwikkelaar, nog steeds met overdraagbaarheid in gedachten. In GNUMail.app gebruikten we het werk van Ludovic en vrienden en hier creëren we een GUI applicatie voor GNUstep en MacOS X.

Er zijn vele tutorials beschikbaar, zowel voor MacOS X als GNUstep. De meeste over GNUstep zijn te vinden op de GNUstep website of op <http://www.gnustep.net>, maar hier zullen we er een paar noemen.

- Een applicatie die gebruik maakt van Gorm en ProjectCenter door Pierre-Yves Rivaille.
- Nicola Pero's tutorial pagina
- Een oudere tutorial over het creëren van een HTMLEditor

<http://stepwise.com/Articles/Technical/HTMLEditor/>

Om meer te leren, kun je ook de broncode lezen, de nib bestanden, enzovoort, van de bestaande GNUstep applicaties (Gorm, ProjectCenter, GNUMail, GWorkspace, enz) en natuurlijk, de gnustep-examples (voorbeelden).

Noem het maar "VerySimpleEditor"

Naast de vele MacOS X tutorials voor InterfaceBuilder die beschikbaar zijn op het net, zullen we de volgende gebruiken als een eerste model:

<http://www.macdevcenter.com/pub/a/mac/2001/05/18/cocoa.html>. De auteur, Mike Beam heeft veel gedegen tutorials geschreven, deze zijn te vinden op <http://www.macdevcenter.com/pub/ct/37>.

Waarom deze? Omdat deze een werkende text editor levert zonder ook maar een regel code te schrijven. Dit demonstreert de kracht van deze ontwikkeltools, of je nu werkt onder MacOS X of onder GNUstep. Met ProjectCenter en Gorm.app onder GNUstep creëren we een erg eenvoudige tekst editor, in staat om te knippen, kopiëren en plakken. Helaas zul je niet in staat zijn je werk op te slaan: vergeet niet dat we geen enkele regel code zullen schrijven. Met ProjectBuilder en InterfaceBuilder onder MacOS X zullen we hetzelfde doen. Uiteraard is er een hoop te verbeteren aan deze editor, dit zullen we aan de lezer overlaten als oefening. Zoals gezegd, dit artikel is geen tutorial!

Daar gaan we.

Onder GNUstep

Open ProjectCenter.app en creëer een nieuw project, met de naam Editor. Kies een Gorm Application project onderin het venster, voordat je de naam opslaat. Dit levert je een Interfaces item in de linker kolom van ProjectCenter.

Een klik op *interfaces* geeft *Editor.gorm*. Dubbelklik op *Editor.gorm* en Gorm.app wordt geopend. Selecteer het default venster (MyWindow) en gebruik de tool inspector om de naam te veranderen in *Editor* in Attributes.

Sleep vanuit het palet een TextView naar het Editor venster. TextView is het grootste object in de geselecteerde palet, te vinden via het meest rechtse icoon bovenin het Palettes venster. Pas de grootte aan om het venster te vullen en je bent klaar.

In de GormInternalViewEditor inspector (terwijl de TextView is geselecteerd), kies *Size* (grootte), en verander de waarden om ze te laten overeenkomen met de Editor venster afmetingen. Deze laatste kunnen op dezelfde manier worden achterhaald, selecteer hiervoor het venster en controleer de grootte in GormNSWindow inspector. Als je de X en Y waarden niet veranderd, bijvoorbeeld, zul je niet in staat zijn de volledige breedte van de editor te gebruiken, of je het venster nu resized of niet.

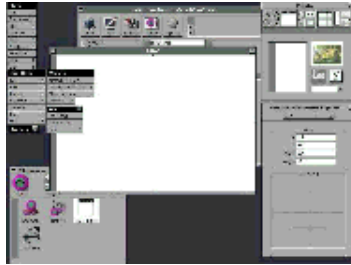
Sla alles op in het Gorm Document menu en sluit deze af om terug te gaan naar ProjectCenter. Selecteer het Build pictogram en klik op het nieuwe build icoon in de tweede horizontale helft van het venster. Alles zou goed moeten gaan als je de juiste voorkeuren hebt opgegeven voor je compiler, debugger, enz. Als je bijvoorbeeld FreeBSD gebruikt, moet je *make* in *gmake* (inclusief pad) veranderen. Dit kan door te klikken op het Settings pictogram van ProjectCenter. Controleer ook de paden in het Preferences menu in ProjectCenter.

Als de compilatie werkte (hij zou het moeten!), doe je hetzelfde met *Run* en je zult de Editor applicatie

zien verschijnen. Speel er even mee, schrijven, knippen, plakken, enz. Uiteraard kun je het later herstarten met het openapp commando.

Hoe lang kostte het? Wel, ik zou zeggen een paar minuten.

Hier is hoe het tijdens de ontwikkelings fase eruit zou moeten zien:

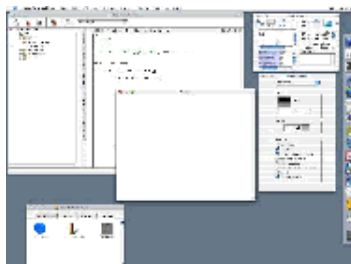


en de resulterende applicatie:



Onder MacOS X

Er is niet veel te zeggen, daar je hetzelfde doet als hierboven. Hier hoe het er uit ziet tijdens het ontwerpen van de GUI:



Hier is de editor "aan het werk":



En nu... een echte werkende applicatie

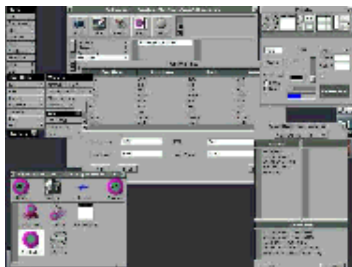
Nu kiezen we een ander voorbeeld van Mike Beam. Deze keer is het een volledig werkende applicatie, in staat om data te beheren: een adresboek. Mike's tutorial over het adresboek (zoals iedere andere) is aan te raden leesstof, om te begrijpen hoe het "ding" werkt. Bekijk ook de tutorial lijst daar Mike verschillende stappen van het ontwikkelingsproces levert voor een en dezelfde applicatie om deze te verbeteren.

En weer bouwen en draaien we de applicatie onder GNUstep en MacOS X.

Onder GNUstep

Net als bij het Editor voorbeeld, start je ProjectCenter.app. Selecteer een Gorm applicatie en noem deze AddressBook. Vanuit Projectcenter start je Gorm door te dubbelklikken in Interfaces -> AdressBook.gorm. Sleep een TableView van het palet naar het default venster. In andere woorden, volg Mike's tutorial als je wilt onder MacOS X. Je zult een paar dingen moeten aanpassen omdat ze verschillend werken in Gorm en InterfaceBuilder.

Bijvoorbeeld, het aantal kolommen in TableView kan niet worden opgegeven vanuit de attributes inspector in Gorm. Om de dingen eenvoudig te houden, kopieer een kolom en plak deze ernaast om het gewilde aantal kolommen te verkrijgen (4 in ons geval. Dit zou iets als het volgende moeten opleveren:



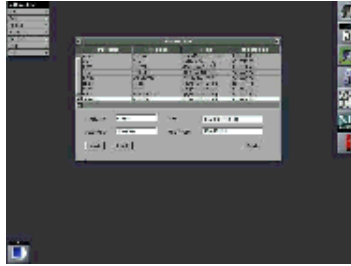
Als je klaar bent, sla je alles op en ga je terug naar ProjectCenter om de code te typen of aanpassen. Voor het geval je wat fouten maakte, Mike levert de volledige applicatie bron code. Als je deze download, is het voldoende om de code te kopiëren en te plakken naar je eigen Controller.m en Controller.h bestanden, zoals gegenereerd door Gorm. *Coca.h* heb je niet nodig daar deze is

gereserveerd voor MacOS X. Overigens zou je het skelet "geschreven" laten door Gorm en een paar dingen aanpassen. Vervang bijvoorbeeld de *void* met *IBAction* in *Controller.h* en *Controller.m*. Voeg *IBOutlet* toe voor het outlets id in *Controller.h*.

Hoe dan ook, als je wilt kun je de volledige geleverde code houden: vervang gewoon de cocoa inclusie met *#import <AppKit/AppKit.h>*. Nu kun je applicatie compileren en uitvoeren.

Daar ben je: je kunt beginnen te spelen met je nieuwe adres boek.

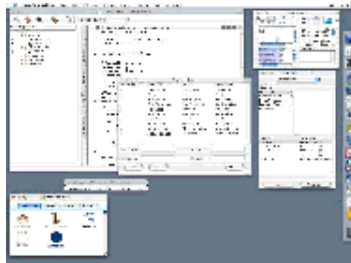
Hier is hoe het eruit ziet:



Onder MacOS X

Mike Beam heeft al het werk al gedaan: wat zou ik nog toe kunnen voegen?

Hier is een screenshot van het ontwikkel proces onder MacOS X:



Hier is de resulterende applicatie:



En toen? Wel, het kostte een paar minuten om de GUI te ontwerpen en de hele code bevat iets meer dan 2000 bytes. Niet slecht he?

Ook hier zijn er nog genoeg verbeteringen aan te brengen, maar het hele ding werkt! Nu kun je met de code gaan spelen om een beter adresboek te krijgen...

GNUstep en MacOS X

Natuurlijk zijn de GNUstep ontwikkel tools nog niet zo ver als Apple's. Apple en NeXT staan voor 15 jaar ervaring met honderden ontwikkelaars. GNUstep is het werk (vrij) van een paar individuen met iets anders te doen voor brood op de plank. Daardoor is het niet verassend om bijvoorbeeld veel meer beschikbare classes te vinden in InterfaceBuilder en in Gorm. Onthoud dat Gorm op versie 0.1.9 zit (of 0.2.0).

Verder hebben we deze testen op de "harde" manier gedaan. We hebben OS X "geport" naar GNUstep. Andersom was eenvoudiger geweest vanwege de bovengenoemde verschillen tussen de tools.

Zo zou het porten van applicaties die ontwikkeld waren onder MacOS X 10.2 veel moeilijker zijn geweest daar de nieuwe Apple ontwikkel tools sterk verbeterd zijn. Zoals al gezegd, zijn er veel nieuwe classes beschikbaar of uitgebreidere.

De tools gaan echter uit van dezelfde filosofie, of ze nu werken onder GNUstep of MacOS X... En GNUstep verbeterd iedere dag. Een ding staat me echt aan: GNUstep mensen werken echt samen. Ze helpen elkaar bij individuele projecten en helpen mee met het verbeteren van de GNUstep kern. Dit is de Free Software manier van werken die ik graag mag. Gefeliciteerd voor dergelijk gedrag Mr Fedor en vrienden.

Stel je voor...

Het doel van dit artikel was om de kracht van de GNUstep "RAD" tools te demonstreren, Gorm.app en ProjectCenter.app. Ondank hun "jeugd" kunnen ze je helpen aardige applicaties te ontwikkelen op een erg eenvoudige manier.

Verder bieden deze tools een erg prettige manier van werken, en tegelijkertijd zijn ze erg efficiënt. Objective C is een erg compacte taal en is, naar mijn mening, makkelijker te leren dan C++ voor iemand met kennis van C (Ik weet het, dit had ik al gezegd!). Dit staat je toe om aardig uitziende applicaties te ontwerpen (ok, het is een kwestie van smaak, maar ik hou van deze look&feel) terwijl ze tegelijk niet al te groot worden.

Ik moet toegeven dat ik nooit ben hersteld van de schok sinds ik voor het eerst kennis maakte met de NeXT machine. Het feit dat Apple een moderne versie van NeXTstep levert charmeert me. Dit is ook waarom ik erg dol ben op projecten als GNUstep of Window Maker. Echter, ik hou van free software en ben geen "fundamentalist" en aansluitend ben ik dan ook niet tegen commerciële software (well, misschien een beetje tegen een specifieke uitgever... maar, alleen een beetje!).

GNUstep kan profiteren van Apple... maar Apple kan ook profiteren van GNUstep. GNUstep is geen Apple concurrent, het is vrije software. Voor zover ik weet, wordt in OS X veel gebruik gemaakt van vrije software. Het kan dus nooit slecht zijn om nog meer vrije software naar Apple te brengen. Wat Ludovic en vrienden hebben gedaan met GNUmail.app is een erg goed voorbeeld van wat er zou kunnen gebeuren.

"Ik had een droom"... Apple leverde de broncode van de meeste van zijn ontwikkel tools aan GNUstep. GNUstep en Apple ontwikkelaars werkten samen om fantastische applicaties te leveren aan Unix gebruikers. En langzaam maar zeker realiseerden de mensen zich dat ze konden leven zonder Windos... Helaas, het maar een droom ;-)

Hoe dan ook, als je GNUstep en diens applicaties niet kent, voel je vrij om ze eens te proberen. Onthoud dat GNUstep een framework is en tools als Gorm en ProjectCenter bieden je alles wat je nodig hebt, om uit te vinden. In andere woorden, met een beetje verbeelding kun je "producten" ontwikkelen die sterk afwijken van wat we vandaag de dag zien: klonen van Windos applicaties!
We leven in een fantastische tijd!

Dank...

Aan de GNUstep mensen: A.Fedor, N.Pero, G.Casamento, P.Y.Rivaille, N.Roard, L.Marcotte, R.Frith-Macdonald, P. C.D.Robert, E.Sersale, A.Froloff, F.Kiefer, M.Viviani, M.Guesdon en degenen die ik nog vergeet voor hun geweldige werk aan het framework of voor de applicaties.

Aan de Window Maker mensen: A.Kojima, D.Pascu en vrienden voor het maken van een vrije NeXTstep interface voor X.

Aan J.M.Hullot en B.Serlet voor het uitvinden van InterfaceBuilder.

Aan "Steve Jobs INC." omdat ze ons NeXT, NeXTstep en MacOS X leverden.

Aan alle mensen hier niet genoemd maar wel hebben meegewerkt aan het minder droevig maken van ons professionele leven.

<p>Site onderhouden door het LinuxFocus editors team © Georges Tarbouriech "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Vertaling info: en --> -- : Georges Tarbouriech <georges.t@linuxfocus.org> en --> nl: Guus Snijders <ghs@linuxfocus.org></p>
--	---