

Diverse organisatie hebben een eigen .NET-ontwikkelstraat. Software Release Magazine was benieuwd naar hoe die er nu zo uitzien. Aanvankelijk waren we van plan op basis van een vragenlijst de ontwikkelstraten van een aantal bekende .NET-partijen op een rij te zetten. Niet iedereen vond het even gemakkelijk die lijst te beantwoorden. Zo ontstond de serie: *De ontwikkelstraat van ... Ditmaal komt Ordina aan bod.*

thema

# De ontwikkelstraat van Ordina

De .NET-ontwikkelstraat van Ordina heet de SMART-Microsoft Software Factory. In het artikel 'Een straat op maat' op pagina XX van dit nummer van Software Release Magazine staat beschreven hoe we tot deze invulling van onze .NET ontwikkelstraat gekomen zijn.

**1. WELKE O/R MAPPER WORDT GEBRUIKT?** Er wordt gebruik gemaakt van NHibernate, de .Net versie van de succesvolle Java Hibernate O/R mapper. Microsoft zelf biedt (nog) geen productieve oplossing voor O/R mapping, derhalve hebben we gekozen voor een open source component.

De NHibernate mapping en alle standaard query's worden geheel gegenereerd uit een van de DSL modellen die we gebruiken, ontwikkelaars moeten alleen de voor de applicatie specifieke query's zelf schrijven.

**2. IN HOEVERRE HOUDT U REKENING MET DE ONTWIKKELINGEN BIJ MICROSOFT OP DIT GEBIED?** We volgen de ontwikkelingen op de voet. Het gebruik van NHibernate is binnen de gekozen architectuur geheel transparant. Dat wil zeggen dat we eenvoudig een andere O/R mapper kunnen inpluggen. We verwachten dat we zodra Microsoft met LINQ komt naast Hibernate ook LINQ zullen ondersteunen. Hierbij kunnen we vanuit de bestaande Business Class DSL de mapping en alle standaard query's genereren. Hierdoor blijven onze modellen, inclusief codegeneratie bruikbaar. De ontwikkelaar zal alleen verschil bemerken bij het schrijven van de applicatiespecifieke query's.

**3. IS ER EEN STANDAARD FRAMEWORK MET KLASSEN WAAROP IEDERE APPLICATIE IS GEBASEERD?** We maken gebruik van een aantal Microsoft componenten zoals

UIPAB en van open source componenten, zoals bijvoorbeeld log4net. Voor zaken die niet door Microsoft of met open source ingevuld konden worden hebben we zelf framework componenten ontwikkeld. De frameworks zijn gestructureerd volgens de lagen in de architectuur en de DSL's. Van alle door onszelf ontwikkelde componenten leveren we uiteraard de source code mee aan onze klanten.

**4. WORDEN KANT EN KLARE MODULES GEBRUIKT? ZO JA, WAARVOOR EN WELKE?** We maken gebruik van een aantal Microsoft componenten zoals UI Process Application Buildingblock en van open source componenten, zoals bijvoorbeeld log4net en NHibernate.

**5. WORDT ER CODE GEGENEREERD EN ZO JA HOE?** Er wordt gebruik gemaakt van Microsoft's Domain Specific Languages (zie vraag 8). Uit ieder DSL model wordt volledig code gegenereerd met behulp van de bijbehorende T4 template engine. Dit betekent dat er een volledig werkende applicatie wordt gegenereerd die exact voldoet aan de Microsoft enterprise architectuur.

**6. MAAKT U GEBRUIK VAN MODELLERING EN IS ER EEN DIRECT VERBAND TUSSEN DE UITEINDELIJKE CODE EN HET MODEL?** Ja en ja. Zie vraag 8.

**7. WERKT U OOK OMGEKEERD?** Nee, de DSL modellen zijn leidend, ze hebben dezelfde status als source-code. Alle code die gegenereerd wordt blijft onaangeraakt door de ontwikkelaar. Daardoor kan er altijd volledig automatisch opnieuw code gegenereerd worden vanuit de DSL modellen. In de door de DSL gegenereerde code is uitgebreid gebruik gemaakt van extension points, punten waarop de

ontwikkelaar de genereerde code kan uitbreiden. Om dit te bereiken maken we uitgebreid gebruik van abstract klassen, virtuele methoden, partial klassen, frameworks. Ook hebben we zelf merge componenten ontwikkeld om handgeschreven delen van onder meer XML-files te kunnen scheiden van de gegenereerde delen.

Alleen als er op basis van bestaande (legacy) code een model nodig is gebruiken we de omgekeerde weg. Dit is wel altijd een eenmalige activiteit, waarna de modellen weer leidend zijn.

**8. GEBRUIKT U DSL'S?** We beschouwen DSL's als de belangrijkste component binnen het software factory concept als het gaat om productiviteitsverhoging. We maken daarom zeer uitgebreid gebruik van DSL's. We hebben vier DSL's ontwikkeld welke gezamenlijk alle onderdelen van de applicatie afdekken. Uit iedere DSL wordt code gegenereerd voor specifieke onderdelen van de architectuur. Uit de combinatie van deze DSL's genereren we een werkende applicatie, geheel onder architectuur.

De DSL's kunnen zowel los van elkaar, maar ook in combinatie met elkaar gebruikt worden. Daarvoor hebben we uitbreidingen op de Microsoft DSL Toolkit ontwikkeld om validatie tussen modellen goed te kunnen doen.

Op het moment dat de eerste release van de DSL Toolkit door Microsoft werd uitgebracht zijn we in productie gegaan met alle DSL's. Ze worden momenteel in verschillende projecten gebruikt.

We hebben direct contact met het DSL team van Microsoft in Cambridge en het Software Factory team in Redmond. Zij zijn zeer geïnteresseerd in onze software factory en de daarvoor ontwikkelde uitbreidingen op de DSL Toolkit.

**9. WELKE IDE WORDT ER GEBRUIKT?** Ordina focust geheel op de Microsoft-technologie. We gebruiken derhalve Visual Studio 2005 met de DSL's als plugins. Verder maken we gebruik van Team Foundation Server.

**10. HOE LANG BESTAAT DE ONTWIKKELSTRAAT IN DEZE VORM EN VERWACHT U BINNENKORT VERANDERINGEN?** SMART-Microsoft is in september 2006 in productie gegaan. In deze straat hebben we alle ervaringen en best practices van de Ordina DaVinci-straat meegenomen en hebben we ons volledig gericht op het software factory concept zoals beschreven in het boek "Software Factories" van Jack Greenfield en anderen. In onze visie is een ontwikkelstraat een levend iets, dat wil zeggen dat deze continu aan verandering onderhevig zal zijn. In de nabije toekomst verwachten we gebruik van de Automation Guidance Toolkit (GAT) van Microsoft en we zien nog uitbreidingen op het gebied van DSL's. Verder houden we nieuwe ontwikkelingen vanuit Microsoft in het oog en zullen deze integreren binnen de software factory. Zie hiervoor ook vraag 2.

## **11. OP WELKE MANIER WORDT GETEST EN WANNEER?**

We maken intensief gebruik van unit tests, welke geïntegreerd zijn in SMART-Microsoft. Ordina ondersteunt daarnaast verschillende methodieken en technieken op testgebied, zoals TMap, ISEB en Testframe. Daarbij zijn we methode-onafhankelijk en wordt de meest logische testaanpak voor een project ingezet.

**12. WELKE TALEN GEBRUIKT U?** Ordina legt de focus op het gebruik van C#. Vanuit software engineering perspectief beschouwen we C# als de beste taal binnen het .NET framework. Dit draagt bij aan een hoge kwaliteit en goede onderhoudbaarheid van de geleverde code.

**13. HOE DWINGEND IS HET GEBRUIK VAN DE STRAAT BINNEN UW ORGANISATIE?** In principe gebruiken alle projecten de straat. Echter, wanneer het type applicatie minder past op de straat wijken we hiervan af. De straat is zo opgezet dat projecten ook eenvoudig onderdelen uit de straat kunnen gebruiken. Zo zal een project dat geen webinterface heeft de hele straat kunnen gebruiken, met uitzondering van de webscenario DSL. Hiervoor gebruiken we dan de Smart Client software-factory van Microsoft.

**14. GEBRUIKT U EEN BEPAALDE METHODE?** We gebruiken al jaren onze eigen agile methode in onze DaVinci-straat. In SMART-Microsoft zijn we overgestapt naar MSF 4 Agile van Microsoft. Deze methode lijkt als twee druppels water op onze eigen methode, maar heeft als grote voordeel dat hij geheel geïntegreerd is met de Team Foundation tools. Hiermee combineren we onze eigen ervaring met de toolondersteuning van Microsoft. Voor de overall projectmanagement methode gebruiken we Prince2. In grotere projecten waarbij samengewerkt wordt met onze Java-groep wordt ook wel de RUP-methode gebruikt.

**15. IS HET GEBRUIK VAN DE ONTWIKKELSTRAAT LOCATIE GEBONDEN?** De SMART-Microsoft Software Factory kan vanuit Ordina-locatie, thuislocatie en klantlocatie worden gebruikt. Ook kunnen klanten gebruik maken van onze ingerichte infrastructuur en tools voor hun eigen projecten. Dit noemen we onze Remote Ontwikkelstraat. Onze klanten krijgen daarmee dezelfde flexibiliteit en snelheid bij het opstarten van projecten als onze Software Factory projecten.

**16. HOE WORDEN MENSEN OPGELEID IN HET WERKEN MET DE ONTWIKKELSTRAAT?** Een ontwikkelstraat is pas effectief als alle projectleden er goed mee om kunnen gaan. Daarom is opleiding een essentieel onderdeel van de ontwikkelstraat. Daarnaast is er een referentie applicatie zodat ontwerpers en ontwikkelaars een voorbeeld hebben. Er zijn opleidingen met de volgende onderwerpen: SMART-Microsoft DSL's & Architectuur, SMART-Microsoft Team System & MSF Agile 4. Naast opleidingen worden de nieuwe medewerkers begeleid door ervaren coaches.