Achtergrond

Volgens Microsoft zou SQL Server Modeling dé nieuwe manier worden om over gegevens te communiceren. Het in 2007 opgestarte project bracht een aantal innovaties met zich mee. De toekomst voor databrowsing en model-gedreven ontwikkeling zag er zonnig uit. Totdat Microsoft in september 2010 besloot de stekker uit het project te trekken.

SQL Server Modeling: nu nog even niet...

Waarom 'Oslo' nooit een succes kon worden

Elger Jonker
is Applicatie ontwikkelaar bij Centric.

Vanaf het begin in 2007 werd voor SQL Server Modeling de codenaam '*Oslo*' gebruikt. Onder deze naam zagen een aantal nieuwe tools het daglicht.

De geschiedenis

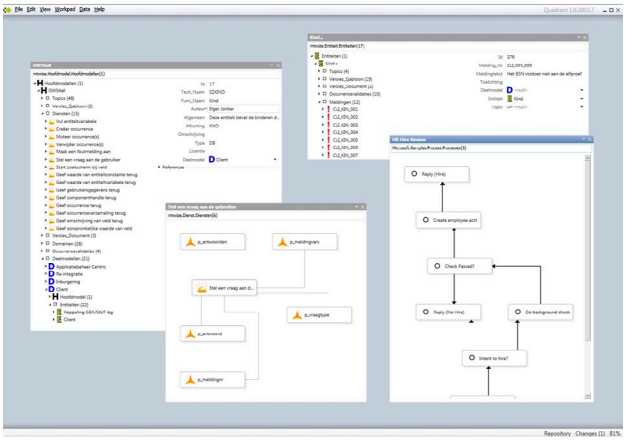
'*Oslo*' werd voor het eerst genoemd in 2007. Toen was er een sterke associatie met Biztalk, maar nog niets tastbaars. Het idee Biztalk werd losgelaten toen Bill Gates tijdens TechEd 2008 de focus legde op applicatiemodellering. Hierna volgde in oktober 2008 de eerste '*Oslo*' Community Technology Preview (CTP). Hierin zaten '*Intellipad*' en de taal '*M*'. Op '*Quadrant*' moesten we nog even wachten, maar het werd wel getoond tijdens de PDC 2008.
Even later begon de communicatiemachine rondom '*Oslo*' pas echt. Op blogs van prominenten als Chris Sells, Douglas Purdy en Kraig Brockschmidt waren artikelen te lezen over waar men mee bezig was. Er was zelfs een aparte blog voor de tool '*Intellipad*', waarin basics en hacks werden gepresenteerd.
De term '*Oslo*' samenvatten bleek een behoorlijke opgave. De eerste tagline kwam van rasoptimist Chris Sells: *"to provide a 10x productivity gain by making model-driven applications mainstream"* (Chris Sells, sellsbrothers.com, 2008).

Het was natuurlijk een streven, een idealistisch statement. Later werd de tagline herschreven naar: "*'Oslo' is the code name for a set of future Microsoft modeling technologies that provide significant productivity gains across the lifecycle of .NET applications by enabling developers, architects, and IT professionals to work together more effectively*" (Oslo Development Center, Augustus 2009).
Eind november 2009 werd de laatste grote update vrijgegeven. '*Quadrant*' kreeg een facelift en de '*M*' grammatica parser was nog stabieler vergeleken met de vorige releases. Daarnaast werd de naam gewijzigd naar '*SQL Server Modeling*'. De naam '*Oslo*' werd volledig in de ban gedaan; het mocht zelfs niet meer gebruikt worden in Microsoft gerelateerde vakbladen. Aan de andere kant was iedereen zo gewend aan de naam, dat het bijna automatisch ging over '*Oslo*' en niet over '*SQL Server Modeling*'.

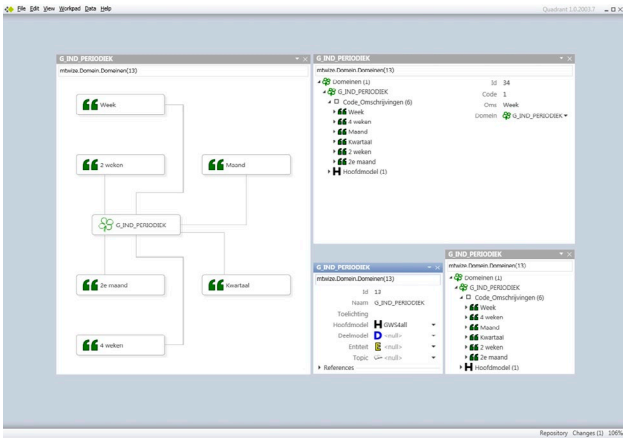
Bij de november 2009 release veranderde ook de homepage van het '*Oslo*'-team. Het werd samengevoegd met het Data-team. De oude homepage, met filmpjes en handleidingen, kwam te vervallen. Op de nieuwe homepage was SQL Server Modeling nog maar een klein kopje. Onder '*Roadmap to the Future*' stond gelukkig nog deze boodschap: "*these technologies introduce new ways of working with SQL Server databases along with greater availability of metadata alongside the data itself*" (Data Development Center, Juni 2010). Het zou een aantal maanden duren voordat er weer genoeg demonstratiefilmpjes en voorbeelden waren.

Figuur 1: SQL Server Modeling Ontwikkelstraat.

Op de voorbeelden na bleef het een jaar nagenoeg stil. Er kwamen enkele updates uit om SQL Server Modeling te



Figuur 2: Quadrant met POC gegevens.



Figuur 3: Views in Quadrant.

kunnen draaien met Visual Studio 2010 en .NET 4.0, maar hierin zaten slechts kleine fixes.
Op 22 september 2010 kwam het bericht van stopzetting. Projectleider Don Box sprak over een verschil tussen de behoefte van de klant en wat SQL Server Modeling beoogde: "*Customers also told us that they wanted the experience to be native to the tools they are already using; specifically either Visual Studio or Microsoft Office.*"
"*Given the increasing adoption of both OData and EDM, we have decided to focus our investments in those technologies to make our modeling vision a reality.*"... "*we will work to make the experience with OData and EDM in Visual Studio and Microsoft Office even better.*" (Don Box, Model Citizen Weblog, 22 Sept 2010)

Wat nu?

Maar wat lijkt Microsoft hiermee nu in de ijskast te zetten? Wat hadden we kunnen verwachten? Waarom was SQL Server Modeling volgens ons de moeite waard?
De ontwikkelstraat volgens SQL Server Modeling
SQL Server Modeling kan op twee verschillende manieren gebruikt worden.

- Om natuurlijke taal toe te passen in .NET applicaties. Hiervoor was de taal '*M*', de toolchain (een M compiler) en '*Intellipad*' nodig.
- Om gegevens te beheren. Ook hierin werd gebruik gemaakt van '*Intellipad*', '*M*' en de toolchain, maar ook de '*Repository*' en '*Quadrant*'.

Bij de laatste manier komen alle tools langs die voor '*Oslo*' werden ontwikkeld.

Codename Intellipad

We beginnen bij het begin: '*Intellipad*'. Dit is een texteditor waarmee domeinspecifieke talen kunnen worden ontwikkeld. Hierin zit een zogenaamde 3-pane modus: een paneel voor '*natuurlijke taal*', een paneel voor de '*grammatica*' en een paneel voor de '*gevonden gegevens in de natuurlijke taal*'. Een domeinspecifieke taal hebben wij niet ontwikkeld; wij werkten in 2-pane modus.
In 2-pane modus kun je werken met de meegeleverde (of zelf ontwikkelde) talen. In de laatste CTP werden drie talen geleverd: een voor SQL, een voor XAML en een voor Entity-Data modellen. Deze hebben allemaal dezelfde invoergrammatica; '*M Types*'.
Een kleine hoeveelheid '*M Types*' levert soms enorm complexe SQL op. Dit komt omdat SQL niet dezelfde constraints kent als '*M Types*'. De gegenereerde SQL is dus niet nodeloos complex.
Met behulp van de toolchain (een set DOS commando's) wordt het '*M Types*' model gecompileerd en de in '*Repository*' geladen. Dit compileren voegt wat extra dimensies toe aan de SQL. Nadat het model in de '*Repository*' is geladen kan deze bekeken worden met '*Quadrant*'.

Codename Quadrant

Het definiëren van talen zal weinig mensen tot de verbeelding spreken, terwijl het voor ontwikkelaars belangrijk is. '*Quadrant*' is een ander verhaal. Het is een indrukwekkende en futuristische manier om door gegevens heen te bladeren. Een

Proof of Concept

Een van de ontwikkelstraten binnen Centric is modelgedreven. Dat betekent dat applicaties worden afgeleid van metadata. Tooling voor deze ontwikkelstraat is grotendeels door onszelf ontwikkeld. Dit kost tijd, kennis en geld. Wanneer een derde partij de ontwikkelstraat aanlevert zou dat schelen.
Van een afstand leek SQL Server Modeling te zijn wat we zochten. Er werd namelijk ook gebruik gemaakt van metadata, model transformaties en domein specifieke talen. Tijdens het proof of concept bleken onze vermoedens grotendeels juist te zijn.
Het proof of concept bestond uit vier fasen.
1. Maken van een datamodel in "M Types" m.b.h. "Intellipad",
2. Importeren van het bestaande data met "M Types" naar de "Repository",
3. Bewerken van gegevens met "Quadrant",
4. Exporteren van bewerkte gegevens naar de modelgedreven ontwikkelstraat.

Elk van deze vier fasen is succesvol afgesloten en SQL Server Modeling kwam als veelbelovend uit de bus. We vonden in onze case meer opportunities dan issues. We hoopte daarom dat in volgende releases de issues opgelost zouden worden.
Tijdens het project hebben we regelmatig contact gehad met het "Oslo" ontwikkelteam via conference calls en e-mails. Dat was erg leuk om te doen. We kregen daarin adviezen hoe e.e.a. aan te pakken. Voor dit soort projecten is dat erg handig. Ook het SQL Server Modeling forum is vaak bezocht.

CTP	Release Date	Contents
October 2008	27 October 2008	SDK, Repository, 3 varianten van de "M" taal
January 2009	30 January 2009	SDK, Repository, 3 varianten van de "M" taal, Excel import tooling
January 2009 refresh	2 March 2009	Kleine updates op de January CTP.
May 2009	26 May 2009	"Quadrant", SDK, unification of "M". No Excel tools
November 2009	November 2009	Quadrant update; betere UI, geïntegreerde M editor. Naam "Oslo" vervalt voor SQL Server Modeling
November R1-2-3	April 2010	Upgrades naar VS 2010 en dergelijke
Final announcement	22 September 2010	Quadrant stopt, M refocust

Tabel 1: Verloop ontwikkeling SQL Server Modeling.

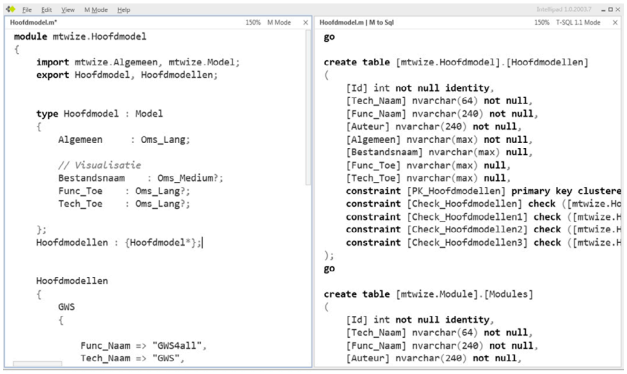
complex model wordt inzichtelijk en beheersbaar. Tijdens ons proof of concept (zie kader), hadden wij een model met meer dan 100 tabellen en 120 relaties. Hieruit een begrijpelijke applicatie maken betekent meestal handwerk, zelfs wanneer applicatie generatoren worden gebruikt. Voor '*Quadrant*' is het echter een kwestie van instellingen aanpassen: het meeste begrijpt de tool al zelf!
Zo worden relaties als een boomstructuur weergegeven, waarbij alleen belangrijke tabellen te zien zijn. De hiërarchie in ons datamodel wordt ook correct weergegeven; met alleen relevante gegevens. Navigeren naar andere gegevens is een kwestie van drag and drop. Sleep een icoon naar de Infinite canvas en details worden getoond. De weergave hiervan kan ingesteld worden als diagram, boomstructuur of als record.

Achter de schermen gebeurt er meer. Het is mogelijk om gegevens af te schermen voor groepen gebruikers. Dat wordt gedaan met Folders, een kolom in een tabel. Dit wordt volledig transparant afgehandeld, wat lijkt op een tabel is in feite een view met daaronder een aantal stored procedures. In die procedures zitten zaken als beveiliging en mapping. Ze worden automatisch gemaakt tijdens het compileren van '*M Types*'.
'*Quadrant*' zelf maakt heftig gebruik van metadata: alle vensters, views, iconen en lijsten zijn opgeslagen in Quadrants eigen database. Zo kan een venster verplaatst worden door de coördinaten ervan in de Quadrant-database aan te passen; het venster verspringt dan. Het aanpassen van Quadrant's look en feel gebeurt vanuit Quadrant. Metadata is het sleutelwoord.

Mogelijke aanpassingen zijn bijvoorbeeld het toevoegen van menu onderdelen. Deze kunnen linken naar eigen assemblies (DLL bestanden). Ook is het mogelijk om met '*M*'-iconen en betere namen in te stellen zoals we zien in de screenshots.

Conclusie

Het 'Oslo'-project was een indrukwekkende bèta. Achter de schermen moet het project nog veel groter zijn geweest. Zo is het de bakermat van de Workflow Engine in .NET 4.0, de .NET service bus en AppFabrik.
Voor ontwikkelaars betekende '*Oslo*' het codewoord voor '*M*', '*Intellipad*' en '*Quadrant*'.
Met de beëindiging van het project is het onduidelijk wat we terug zullen zien. Volgens een van de projectleiders is het de bedoeling dat de gemaakte stappen terugkomen in andere toepassingen. Wat of wanneer kon hij helaas niet zeggen.
Wat we wel zeker weten is dat '*M*' in ontwikkeling blijft; dit staat in het afsluitende statement van Don Box. Dat is zeker goed nieuws gezien de veelzijdigheid en eenvoud van de taal. We misten de mogelijkheid om gegevens te updaten, dus het zou zo kunnen dat dit straks wel mogelijk is.
Wat er met '*Quadrant*' gaat gebeuren is een raadsel. SQL Server Management studio zou zeker kunnen profiteren van een nieuwe gegevensbrowser. '*Quadrant*' heeft geweldige ondersteuning voor relaties en navigatie. Even door de database heen fietsen wordt kinderspel. Vergelijk dit met de wat sobere tegenhangers in Management studio: Database Diagrams of Edit Top 200 Rows. Verder zou '*Quadrant*', met wat aanpassingen, een beter alternatief vormen voor het beheren van database diagrammen.



Figuur 4: Intellipad in 2 pane mode, met POC gegevens.

Omdat '*Quadrant*' een CTP release was ontbrak er nog een aantal dingen. Zo zou een zoekfunctie, ondersteuning voor wijzigingshistorie (CDC) en gebruik van eigen datatypes veel kunnen toevoegen.
Samenvattend is met SQL Server Modeling een indrukwekkende en vernieuwende set tools neergezet. De mogelijkheid om een eigen grammatica te maken en eenvoudig door grote datasets heen te bladeren vonden we erg aantrekkelijk. We hopen in de toekomst '*M*' en '*Quadrant*' in een aangepaste vorm terug te zien. Beide zullen een behoorlijke impact hebben op hoe we met gegevens omgaan.
Inmiddels zijn alle documenten, filmpjes en websites rondom deze techniek deprecated verklaard. De pagina's beginnen met de melding "*This content is no longer valid. For the latest information on "M", "Quadrant", SQL Server Modeling Services, and the Repository, see the Model Citizen blog.*]". «

Referenties

- Microsoft Data Development Technologies: Past, Present, and Future, Kraig Brockschmidt, April 2010, http://msdn.microsoft.com/library/ee730343.aspx
- Design Time Code Generation and Runtime Model-Driven Generation, Rockford Lhotka, Magenic, April 2010, http://msdn.microsoft.com/en-us/library/ff621668.aspx