



Databases zullen veranderen om rules te kunnen faciliteren

De connectie tussen Business Rules en Databases

Malcolm Chisholm

Wat zijn Business Rules precies en waar ligt het raakvlak met het managen van data? Op die vragen geeft dit artikel een helder en duidelijk antwoord. De clue geven we vast weg: het concept is zeer relevant voor datamanagement en gaat zeker impact hebben op de database-structuur.

In de informatietechnologie wordt de term 'Business Rules' vaak gebruikt en de populariteit ervan groeit. Meestal is het een generieke manier om de proceslogica te beschrijven die programmeurs al decennia lang toepassen.

Het staat buiten kijf dat traditionele proceslogica in enigerlei mate 'Business Rules' bevat, maar door het simpel 'Business Rules' te noemen vertelt het ons niets nieuws. Gebruikers uit de business en IT-managers beginnen zich dat nu te realiseren.

Om heel goede redenen houden ze er niet van om termen te gebruiken die óf alleen maar een nieuwe naam zijn voor iets dat allang bestond, óf die geen enkel onderliggend concept hebben. Dat is jammer, omdat er onder Business Rules een echt fundament ligt; bovendien kan een Business Rules aanpak in de IT enorme voordelen opleveren.

Ik vind het atomaire principe buitengewoon belangrijk bij Business Rules

Een tweede probleem is dat Business Rules blijkbaar alleen refereren aan logica, terwijl op dit moment de echte problemen in de IT te maken hebben met data en architectuur. Als Business Rules alleen maar met programmeren te maken zouden hebben, zullen ze ons niet kunnen helpen bij het oplossen van problemen in data en architectuur. De werkelijkheid is echter dat er een nauwe band bestaat tussen Business Rules en data. Architectuur bestaat op een ander niveau en hoewel ook architectuur een relatie met Business Rules heeft, concentreert dit artikel zich op de connectie met databases.

We zien ons derhalve geconfronteerd met twee directe uitdagingen: wat zijn Business Rules nu eigenlijk precies; en hebben Business Rules enige relevantie voor het managen van data? Ik herhaal nog maar eens dat Business Rules een valide concept representeren en dat dat concept zeer relevant is voor datamanagement en derhalve voor databases.

Wat zijn Business Rules?

The Business Rules Group (www.businessrulesgroup.org/defnbrg.shtml) heeft een belangrijke rol gespeeld bij het onder de aandacht brengen van Business Rules bij IT, en daarom verdienen de definities die zij geven voor de term 'Business Rules' aandacht. Ze hebben twee definities:

- vanuit business perspectief, "A Business Rule is guidance that there is an obligation concerning conduct, action, practice, or procedure within a particular activity or sphere";
- vanuit IT-perspectief, "A Business Rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business".

Persoonlijk vind ik deze definities te algemeen. Bovendien kunnen ze worden geïnterpreteerd op een manier die niet uitvoerbaar is. Zo deed ik bijvoorbeeld eens een Business Rules project voor de 'private client' divisie van een groot effectenkantoor in New York. Een 'private client' is een rijk persoon aan wie het is toegestaan om rechtstreeks investeringsbeslissingen te nemen over het geld dat hij bij de effectenmakelaar heeft ondergebracht. Ergo, deze klanten zouden 'Business Rules' aan de makelaar kunnen dicteren. Toen ik naar de lijst van Business Rules keek die het effectenkantoor had vastgelegd, vond ik bijvoorbeeld "Deze portfolio moet een jaarlijkse opbrengst van minimaal 10 procent opleveren".

Dat lijkt misschien een Business Rule, maar in werkelijkheid is het een business doelstelling. We kunnen op geen enkele manier programmeercode schrijven die een opbrengst van 10 procent op een investering garandeert. Als dat kon, zouden we immers allemaal rijk zijn.

Vanuit mijn eigen ervaring weet ik dat Business Rules een aantal karakteristieken moeten hebben:

- Alle Business Rules zijn óf atomaire, óf zijn samengesteld uit andere Business Rules. Een atomaire Business Rule kan niet uiteenvallen in meerdere Business Rules. Een niet-atomaire Business Rule bestaat juist alleen uit andere Business Rules met een enkelvoudige onderlinge relatie.
- De termen in een Business Rule representeren brondata en eenvoudige activiteiten. De brondata moeten zijn opgeslagen in een database. In de praktijk is dat niet altijd het geval, maar dat zou wel zo moeten zijn.
- Als een Business Rule wordt uitgevoerd genereert het een resultaat dat kan worden opgeslagen in een enkele database-kolom.

Ik vind dat het atomaire principe buitengewoon belangrijk is bij Business Rules. Traditionele, handgeschreven programmeercode wordt vaak beschreven als 'procedureel', zo suggererend dat het niet consistent is met Business Rules. Daar ben ik het niet mee eens. Business Rules kunnen inderdaad een volgorde van stappen specificeren, waarop bedoeld wordt met de term 'procedureel'. Desalniettemin stelt met de hand vervaardigde programmeercode een Business Rules aanpak wel degelijk voor echte problemen, zoals:

- Het is erg moeilijk om uit te vinden waar de ene Business Rule eindigt en de andere begint;
- Omdat de Business Rules herhaaldelijk gespecificeerd worden op verschillende plaatsen, kunnen ze niet eenvoudig worden hergebruikt en gemanaged. Als een rule dus verandert, moet hij op diverse plaatsen worden aangepast;
- Business Rules zijn in lange sequenties aaneen geschakeld en er worden geen tussentijdse resultaten opgeslagen, een beperking van de informatie die een applicatie zou kunnen geven. Dit heeft ook negatieve gevolgen voor impact assessment en control.

Inductie versus Column Updates

Het idee dat een Business Rule atomaire is correspondeert met het idee dat een Business Rule in een enkele database-kolom zou moeten zitten. De datamodelleer-rule over atomiciteit bepaalt dat één database-kolom maar één enkele definitie mag hebben. Dus als de definitie een berekening of een afleiding is – in tegenstelling tot een karakteristiek of waarde die in de echte wereld wordt gemeten of waargenomen – dan zit er één enkele Business Rule achter deze database-kolom.

Niet alle Business Rules applicaties volgen dit patroon expliciet. Een groot aantal van dergelijke applicaties houdt zich bezig met gevolgtrekkingen, inferentie. Dat is een logische aanpak om

een conclusie te kunnen trekken op basis van een verzameling feiten. We kunnen bijvoorbeeld een Business Rules applicatie hebben die kredietapplicaties evalueert. Op de informatie in een applicatie voor kredietaanvragen zijn vele Business Rules van toepassing. Ze schakelen aaneen en uiteindelijk levert dat als resultaat op of het krediet al dan niet verstrekt wordt (of misschien zelfs de aanvraag wordt teruggestuurd met het verzoek om extra informatie).

Het gebruiken van Business Rules om gevolgtrekkingen te maken is een belangrijke categorie met een lange historie. Het is echter nog steeds geen gemeengoed, in plaats daarvan hebben de meeste ondernemingen grote aantallen eenvoudige berekeningen, afleidingen, en zaken van voorwaardelijke logica. Die kunnen elk op zich misschien wel eenvoudig zijn, in geaggregeerde vorm kunnen ze overweldigend complex zijn. Het is dit gebied, eerder dan inductieve logica, waarin Business Rules hun grootste impact hebben.

Helaas beklemtoont veel van de literatuur over Business Rules het gebruik in inductie en gaat voorbij aan het berekenen van waarden van database-kolommen. Dat vertroebelt de relatie tussen Business Rules en data; ook is het moeilijker om te zien hoe de Business Rules aanpak waardevol kan zijn in normale computerapplicaties.

Regels, Taal en Attributen

Een ander probleem dat de connectie tussen Business Rules en databases verbergt is dat er in het algemeen aan wordt vastgehouden om de Business Rules in termen van natuurlijke taal te formuleren. Het is een erg sterke trend onder praktikanten van Business Rules om te blijven proberen een ondubbelzinnige manier te vinden om Business Rules in menselijke talen te formuleren. Persoonlijk heb ik grote twijfels over deze aanpak. Elk gebruik van menselijke taal heeft de neiging om in mindere of meerdere mate af te hangen van arbitraire conventies binnen een taal. Om verwarring te zaaien verschilt het Engels dat in Engeland gesproken wordt aanzienlijk van het Engels dat men in Amerika spreekt. Ondernemingen zitten vol termen die op de

BRPN

Ook in Nederland is het Business Rules Platform erg actief. Het BRPN richt zich op Business Rules Management (BRM). BRM heeft betrekking op het expliciet, en daarmee beheersbaar, maken van bedrijfsregels, zodat zeker kan worden gesteld dat deze daadwerkelijk worden toegepast in de bedrijfspraktijk door zowel medewerkers als geautomatiseerde systemen. Het BRPN heeft als doel de bekendheid met BRM in de Nederlandse markt te bevorderen, door objectieve uitwisseling van kennis en ervaring tussen gebruikers, leveranciers en wetenschap te bewerkstelligen.

Voor meer informatie: www.brpn.org

ene manier in het ene bedrijfsdeel gebruikt worden, en op een andere manier in een ander. Bij de marketing-afdeling bijvoorbeeld, gebruikt men voor prospects ook het woord 'Klant', maar bij de debiteurenadministratie kunnen dat alleen individuen zijn die het bedrijf geld schuldig zijn, of dat in het verleden waren. Dit legt grote beperkingen op aan de manier waarop we op een accurate manier Business Rules kunnen voorstellen in termen van natuurlijke taal.

De pogingen om Business Rules te formuleren in natuurlijke taal staan in scherp contrast met de behoefte van datamodelleerders om goede definities voor attributen en kolommen te creëren.

Als we de tabellen van een willekeurige databases bij een willekeurig bedrijf bekijken zullen we constateren dat vele daarvan afgeleid of berekend zijn, en niet dat er iets wordt opgeslagen dat kan worden geobserveerd of gemeten inzake de entiteiten die ze representeren. Dat zijn wel de kolommen die de uitkomsten van atomaire Business Rules kunnen vastleggen.

Deze kolommen (of attributen) hebben echter meestal erg vage definities. Zo kan het attribuut Order Total in een Order-tabel bijvoorbeeld gedefinieerd zijn als 'Het totale bedrag van de order'. Het zou ook kunnen worden gerepresenteerd door een Business Rule ongeveer als deze:

```
Order.Order Total =  
    (Sum of the Order Lines.Amount) -  
    (Order.Order Discount Amount) +  
    (Order.Order Tax Amount)
```

Uit dit voorbeeld blijkt dat de definitie vrijwel niets van de logica van de Business Rule bevat en dat er niet wordt gerefereerd aan de kolommen die de rule samenstellen. Eerlijk gezegd zijn attribuutdefinities in datamodellen meestal algemene statements van lage kwaliteit. Ze zijn niet onjuist – wie kan het niet eens zijn met 'Het totale bedrag van de order'. Het probleem met deze definitie is dat hij geen werkelijk bestaande specificering bevat. Als we dat vergelijken met de corresponderende Business Rule definitie, zien we direct het verschil en realiseren we ons wat er mist.

De consequentie is dat de adoptie van de Business Rules aanpak gevolgen heeft voor datamodellering. De rules moeten tot in detail worden beschreven en kunnen niet worden vervangen door vage definities.

Attribuut-inflatie

Er is een dieper effect dat dit probleem veroorzaakt. Toen ik aan mijn eerste Business Rules projecten begon, was ik verbijsterd over de enorme toename van het aantal kolommen in de databases waar ik mee te maken had. In een bankapplicatie kunnen we bijvoorbeeld de tabel *Account Monthly Details* hebben en dat zou een attribuut *Account Monthly Fees* kunnen hebben met als definitie 'The amount of fees assessed per month for the account'. Opnieuw zeg ik: daar zal iedereen het mee eens zijn. Maar stel nu eens dat er drie typen Account zijn: Individual, Corporate en

Not for Profit, met de volgende eigenschappen:

- Er worden geen fees berekend op een Not for Profit Account;
- Er wordt een fee van 0,005 procent geheven op het gemiddelde dagelijkse saldo van een Corporate Account;
- Er wordt een fee geheven van 10 dollar per maand op Individual Accounts met een saldo minder dan 1.000 dollar.

Wat we eigenlijk hebben, zijn drie atomaire Business Rules.

Dat betekent dat we echt drie attributen nodig hebben om de resultaten van elke Business Rule op te kunnen slaan, in plaats van Account Monthly Fees. Dat zouden zijn:

- Not for Profit Account Monthly Fees;
- Corporate Account Monthly Fees;
- Individual Account Monthly Fees.

Toegegeven, de Not for Profit Account Monthly Fees zouden nu een waarde van nul opslaan, maar in de toekomst zouden we de rule zodanig kunnen wijzigen dat wel een maandelijke fee wordt geheven. Deze toename van het aantal attributen is een gevolg van de hoge mate van precisie die Business Rules vereisen, vergeleken met de veel minder exacte definities die meestal in datamodellen worden gebruikt. Elke onderneming die de Business Rules aanpak voorstaat kan er dus op rekenen dat het aantal kolommen in de database aanzienlijk zal groeien.

Als we in dit voorbeeld geen Business Rules hadden gebruikt, zouden de programmeurs op de hoogte moeten zijn van de rules en er naar moeten handelen. Als de eis dus zou zijn het geven van het totaal van alle fees op Corporate Accounts per maand, moeten alle waarden in Account Monthly Fees die betrekking hebben op Corporate Accounts worden opgeteld. Met de Business Rules aanpak, die ons dwingt een kolom Corporate Account Monthly Fees te creëren, hoeven we alleen de waarden in deze kolom te totaliseren. Business Rules kunnen dus een aanzienlijke impact hebben op database-structuren.

Opslag van tussentijdse resultaten

Een ander gevolg van de Business Rules aanpak is de behoefte om tussentijdse uitkomsten op te slaan. Het atomair maken van Business Rules vereist dat meestal. Stel dat we de volgende logica hebben:

$$A = (B + C + D) - (E + F).$$

In een Business Rules aanpak moeten we additionele kolommen creëren om te voldoen aan de atomiciteit van de rules:

$$X = B + C + D$$

en

$$Y = E + F.$$

Vervolgens wordt de oorspronkelijke rule:

$$A = X - Y.$$

In handmatig geprogrammeerde code kunnen programmeurs deze situatie beheersen door herhaaldelijk X en Y te berekenen in programmalogica zonder de resultaten op te slaan. De waarden worden vastgehouden in het geheugen en gevoed aan

andere Business Rules, of output op schermen en in rapportages. Meestal betekent dat dat de logica redundant gerepliceerd moet worden overal waar de waarden nodig zijn. Toegegeven, de programmalogica zou gecentraliseerd kunnen worden, maar dat wordt zelden gedaan voor dergelijke tussentijdse uitkomsten. Het is ook onmogelijk om logica te centraliseren als de data over verschillende applicaties lopen – de data moeten dan redundant gerepliceerd worden in deze applicaties. Tussentijdse resultaten zijn vaak moeilijk te herkennen voor een datamodellerder, maar ze zijn vereist in een Business Rules aanpak en nogmaals: ze leiden tot kolom-inflatie in de database.

De negatieve kant van Business Rules

Er is een probleem dat nauw samenhangt met tussentijdse uitkomsten. Het speelt een belangrijke rol bij de negatieve kant van Business Rules: het ontbreken van programmeursupport bij de automatisering van Business Rules. In een traditioneel software-ontwikkelingsproject voeren programmeurs logica met de hand in en voeren vergaande testen uit. Ze worden geholpen door tools voor het debuggen van programmeercode. Zulke tools zijn ook handig als zich een productieprobleem voordoet waarbij datawaarden geproduceerd door Business Rules worden bevestigd. Als een gebruiker echter de rules definieert in een Business Rules application kan er niet even een programmeur worden geroepen als de gebruiker een twijfelachtige uitkomst in een rapport of op het scherm ziet staan. In zo'n situatie moet de gebruiker proberen uit te vinden welke Business Rule in de keten van de desbetreffende Business Rules niet werkt zoals verwacht. Zonder een goede verzameling tussentijdse resultaten wordt de gebruiker gedwongen ingewikkelde Business Rules te modelleren, hetgeen moeilijk is, misschien zelfs onmogelijk. Het is veel eenvoudiger om door een complete verzameling tussentijdse resultaten te kijken om er achter te komen waar het potentiële probleem zich voordoet. Dit is ook van toepassing op het gebruik van Business Rules bij het maken van gevolgtrekkingen, dat ik eerder noemde. Het kan ongelooflijk moeilijk zijn uit te vinden hoe een gevolgtrekking tot stand is gekomen, zonder het opslaan van tussentijdse uitkomsten van elke Business Rule die betrokken was bij het maken van de gevolgtrekking. Er kunnen zelfs juridische redenen zijn om dit te doen als de conclusie aanzienlijke impact heeft op een individu of onderneming – zoals bijvoorbeeld het afwijzen van een krediet. De mogelijkheid tot het uitvoeren van een audit op de executie van Business Rules vereist opslag van de tussentijdse resultaten en heeft invloed op het aantal kolommen nodig in de database.

Business Rules en Metadata

Tot nu toe hebben we het gehad over Business Rules in termen van de bronkolommen die ze gebruiken en de doelkolommen waarin ze ieder resultaat dat ze produceren opslaan. Zoals eerder gesteld is natuurlijke taal niet ideaal voor het definiëren van Business Rules. Business Rules bestaan uit termen en relaties

tussen die termen, wat betekent dat ze kunnen worden uitgedrukt in datamodellen van metadata. Hierdoor kunnen Business Rules eenvoudig worden gedeclareerd.

We nemen als voorbeeld een rule voor het berekenen van een percentage. Het heeft de volgende kenmerken: een numerator; een denominator; precisie bij het afronden; en misschien de behoefte om te weten of de denominator de som is van wat we tonen als de numerator en denominator – bijvoorbeeld: $(N/D) \times 100$ versus $(N/(N+D)) \times 100$.

Men kan zich gemakkelijk een scherm voorstellen waar deze vier informatie-items kunnen worden ingevoerd, samen met de mogelijkheid om de kolom te selecteren waar het resultaat opgeslagen moet worden. Het is ook gemakkelijk een meta-model voor deze Business Rule voor u te zien.

Daarom moet het database-ontwerp nu ook metamodelen bevatten voor alle klassen Business Rules die gedefinieerd moeten worden. Dat brengt een veel rijkere verzameling metadata direct in de productie-databases. Het is een behoorlijk lastige aanpak om de metadata in een speciale repository te bewaren, apart van alle operationele productie-databases. Zo'n repository bevat meestal metadata in de vorm van documentatie en dat verschilt nogal van de metadata die nodig zijn om de Business Rules te automatiseren.

Natuurlijk zijn er alternatieve ontwerpen en architecturen voor de implementatie van geautomatiseerde Business Rules. Toch maakt het invoeren van Business Rules metadata in de productie-database het mogelijk om rijkere metadata te definiëren. Rules kunnen bijvoorbeeld assigned classifications zijn. Met een metamodel voor de definitie van een fee-rule kunnen we de fee categoriseren als managementfee, operational fee of performance fee, of welk type fee dan ook. Governance metadata kunnen ook worden toegevoegd aan de rule-metamodelen, hetgeen erg goed van pas kan komen in een productie-database.

Tot besluit

Zo zien we dat er vele connecties bestaan op vele niveaus tussen Business Rules en databases. Als de Business Rules aanpak meer gemeengoed wordt kunnen we verwachten dat databases zullen veranderen om het te faciliteren. Het resultaat is waarschijnlijk een grotere mate van precisie, eenvoudiger onderhoud en rijkere functionaliteit voor de applicaties die gebaseerd zijn op deze databases.

Meer over Business Rules kunt u vinden bij The Business Rules Group: www.businessrulesgroup.org/defnbrg.shtml

Dit is een bewerkte en vertaalde versie van de originele Engelse tekst, die u kunt vinden op onze website www.dbm.nl onder 'speciaal', 'extra materiaal'. In geval van discussies is de originele Engelse tekst doorslaggevend.

Malcolm Chisholm is directeur van Askget.com Inc te New Jersey.