

Woningcorporaties in Nederland hanteren aanzienlijke budgetten voor het planmatig onderhoud van woningcomplexen. Denk aan het periodiek schilderen van houtwerk, vervangen van dakdelen, installeren van liften, ketels en dubbelglas, en het repareren van goten en houtrot.

Planmatig onderhoud bij woningcorporatie

USoft bouwde beheerapplicatie voor Vestia

Er is veel geld te verdienen door precies op het juiste moment op de juiste plekken in te grijpen. Dan is offertes vergelijken niet genoeg. Je moet zelf gedetailleerde kennis opbouwen over je woningbezit.

USoft bouwde in nauwe samenwerking met woningcorporatie Vestia Groep een applicatie om de kosten van planmatig onderhoud aan woningcomplexen te plannen, te onderhouden en te beheren. De applicatie is nu drie jaar oud en in gebruik bij twee grote woningcorporaties.

In dit maatwerkproject werd een nieuwe ontwikkelmethodiek toegepast. Er werd geen object model gemaakt of een andere architectuur gericht op het organiseren en beheersen van geprogrammeerde code. In plaats daarvan werden agile ontwikkelen, 4GL prototyping, SBVR semantiek en een USoft rules engine met elkaar gecombineerd. Laten we deze aanpak met een geïmproviseerd etiket kortweg 'Agile 4GL+' noemen.

Business rules, de OMG standaard SBVR voor eenduidig formuleren van business logica, en agile staan sterk in de belangstelling, maar worden meestal alleen in verband gebracht met organisatiebrede, analytische applicaties: BI, BPM, OLAP, data mining, decision support of datawarehousing. Ook bij typische business rule thema's zoals governance en compliance wordt vooral gedacht aan de besturing van grote corporaties of aan de juridische of financiële verantwoording van een heel bedrijf.

Dit project illustreert daarentegen hoe succesvol agile en business rules kunnen zijn in combinatie met elkaar en in de context van een kortlopend project. De doelapplicatie is weliswaar data-intensief

en biedt deels OLAP functionaliteit, maar het is meer een totaaloplossing voor een deelgebied (planmatig onderhoud) dan een analytische tool.

Nieuwe manier ontwikkelen

Om deze applicatie te bouwen werd een centrale lijst (een rulebase) met elementaire specificaties in natuurlijke taal aangelegd, die één voor één op correctheid werden gecheckt door de opdrachtgever, en vervolgens één voor één door USoft werden geïmplementeerd.

De implementaties werden gebouwd in een USoft rules engine. In deze omgeving wordt de volledige bedrijfslogica gerealiseerd door een relationeel datamodel op te stellen en daar SQL statements "tegenaan" te schrijven die bedrijfsregels uitvoeren. Het gaat om een 4GL RAD omgeving, dus op basis van deze declaratieve implementaties produceert de tool meteen bruikbare applicatie prototypen. Het gebruik van USoft, SQL of zelfs relationeel is niet de essentie van deze aanpak. Het data model had een UML model kunnen zijn en de SQL had object-geïntegreerde code kunnen zijn. Voorbeeld 1 vat de aanpak samen.

Specificaties

De specificaties zijn elementaire beweringen in natuurlijke taal, in dit geval het Nederlands. Ze voldoen aan de OMG standaard SBVR (Semantics of Business Vocabulary and Rules) voor het gestructureerd opstellen van business vocabulaire en bedrijfsregels.

We tekenden enkele diagrammen ter ondersteuning, maar beslissend waren altijd de natuurlijke-taal beweringen zelf. Toch werden er geen documenten



Rob van Haarst
Senior Consultant
bij USoft b.v.

geschreven met een inhoudsopgave, hoofdstukindexing en index. Elementaire beweringen staan juist op zichzelf. Ze kunnen onafhankelijk van elkaar gevalideerd worden en ongeordend worden opgeslagen in een meta-data tabel.

In de praktijk bleken veel betrokkenen de vertrouwde document structuur te missen. Het kost tijd voordat iedereen om kan gaan met 'losse' taalbeweringen die geheel voor zichzelf spreken en niet in een vaste ordening worden aangeboden.

Maar daar staan wel heel veel voordelen tegenover. De opdrachtgever en haar partners (zoals onderaannemers en toeleveranciers) begrijpen onmiddellijk wat er van hen verwacht wordt als ze per natuurlijke-taal bewering moeten valideren of die bewering klopt en gewenst is, ook als dat schriftelijk moet omdat er geen vergadertijd is. Ze hoeven zich niet eerst een notatie of diagramtechniek eigen te maken. Specificaties kunnen volledig in het jargon van de klant worden opgesteld. Er zijn geen aparte functionele en technische ontwerpdocumenten, waardoor je sneller kunt werken.

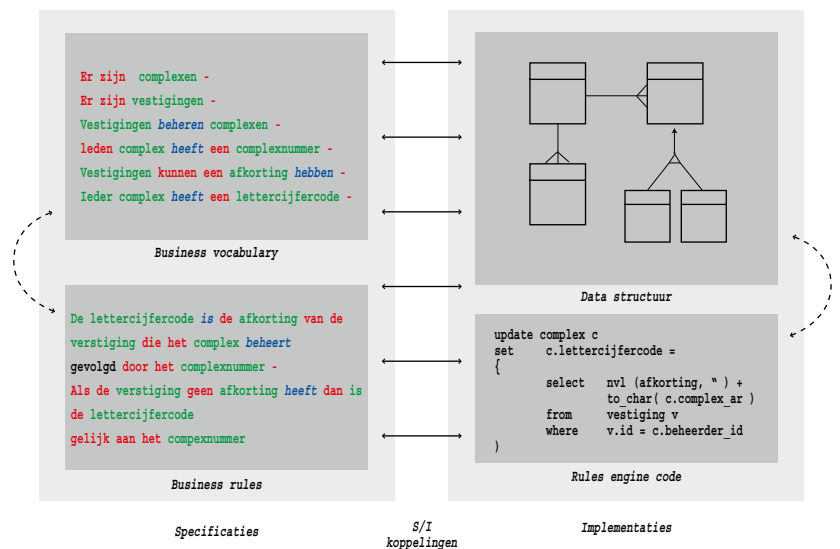
De ongeordendheid van beweringen biedt ook voordelen. De tool hyperlinkt automatisch ieder gebruik van een term naar de definitie(s) van die term. Dat werkt omdat die definitie voor zichzelf spreekt, en niet deels steunt op de context van een specifieke plaats in een document. Er kan tooling gebruikt worden om automatisch beweringen te groeperen die dezelfde term gebruiken. Daardoor komt veel sneller aan het licht dat een term op verschillende plaatsen niet met dezelfde betekenis wordt gebruikt. Elke bewering is een meta-data record waarin relevante kenmerken zoals datum aangemaakt, laatst gewijzigd en door wie, applicatie versie, prioriteit, gevalideerd ja/nee, reden opname, en een classificatie naar onderwerp kunnen worden bijgehouden. Dit biedt de mogelijkheid om met ad-hoc queries vele nuttige dwarsdoorsneden te maken.

Het meest spectaculaire voordeel van deze vorm van specificatie ligt echter in de gelijkvormigheid tussen specificaties en implementaties, waardoor koppelingen tussen de twee gemakkelijk gemaakt en onderhouden kunnen worden.

Implementaties

De software werd geïmplementeerd door een relationeel model met bedrijfsregels. Het model bevat tabellen, ouder-kind relaties en generalisatie-specialisatie relaties. Het biedt data opslag voor prototypen en de basisstructuur voor de bedrijfsregels. De bedrijfsregels worden uitgedrukt in SQL statements met slechts hier en daar (drie procent) exits naar geprogrammeerde componenten, en uitgevoerd door een rules engine.

Het data model behoeft weinig toelichting. Deze methodiek leidt in de praktijk tot een bepaalde manier van modelleren, waarin ook afgeleide gegevens veelal worden uitgemodelleerd. De analyse



Voorbeeld 1: Gekoppelde specificaties en implementaties.

van Malcolm Chisholm in Database Magazine dat goed gebruik van business rules leidt tot datamodellen met veel meer kolommen dan voorheen gebruikelijk, wordt hierdoor bevestigd. In Voorbeeld 1 hierboven moet bijvoorbeeld niet teruggeschrokken worden voor het modelleren en opslaan van lettercijfercodes, ook al zijn die afleidbaar uit andere opgeslagen gegevens.

Een rules engine zoals ILOG, Corticon, PegaSystems, USoft is een beslismachine die in staat is om automatisch bedrijfsregels met elkaar in verband te brengen en de executie van samenhangende regels op de juiste manier aan elkaar te rijgen.

In de USoft implementatie wordt dit bereikt door een bedrijfsregel te beschouwen als een datasamenhang die wordt gedeclareerd door een SQL statement. In voorbeeld 1 definieert een SQL statement hoe de lettercijfercode van een wooncomplex samenhangt met andere data. De rules engine garandeert dat die samenhang behouden blijft, ongeacht welk bedrijfsproces de relevante data "raakt". De applicatie is zeer wendbaar: bestaande gegevens kunnen elegant aan een nieuwe regel worden aangepast, door er hetzelfde SQL statement op los te laten dat ook al dient voor de nieuwe bedrijfsregel zelf.

Het gaat hier echter niet zozeer om de specifieke kenmerken van de rules engine die in dit project is gebruikt, maar wel om het algemene principe dat rules engines ervoor kunnen zorgen dat business rules onafhankelijk van elkaar maar ook in de juiste samenhang geëxecuteerd worden. Dit maakt het mogelijk om de elementaire beweringen uit het naar SBVR opgestelde conceptuele model, die immers óók alle afzonderlijk en in samenhang waar zijn, één op één te koppelen aan implementaties. Als een regel wordt gewijzigd of gedropt uit het kennismodel is onmiddellijk duidelijk welke implementatie

Ongeordendheid van beweringen biedt ook voordelen.

**‘Agile 4GL+’
is gericht
op gestructureerd
opslaan van
meta-data en
applicatie
data.**

Nederlandse investeerders kopen USoft

Investeerders Patrick Dekker en Egbert Hendriks hebben in januari van dit jaar USoft gekocht van Ness Technologies. De software van USoft is van Nederlandse origine. De oorsprong gaat terug tot 1986, toen het bedrijf TopsySystems werd opgericht dat een ontwikkeltool in de markt zette met een strikte scheiding tussen bedrijfsregels en applicatie. Unisys liet in 1995 zijn oog vallen op dit bedrijf en noemde het na de overname USoft. Unisys verkocht USoft in 1998 aan de Israëlische softwareleverancier Advanced Technology (ATL) dat op zijn beurt in 1999 werd overgenomen door Ness Technologies.

De nieuwe eigenaren, Patrick Dekker en Egbert Hendriks, zijn tevens de oprichters/eigenaren van HintTech, een Delftse leverancier van weboplossingen op basis van Tridion-software. ‘Het duo was op zoek naar een nieuwe investering. Ness Benelux stond in de etalage’, vertelt directeur Crétien Jung. ‘Het is geen overname door HintTech. USoft blijft een gescheiden organisatie met een eigen productacquisitie. Onze software voor data-intensieve transactieverwerking is heel wat anders dan de websystemen van HintTech. Wel wordt gekeken of beide partijen elkaar kunnen ondersteunen. Zo is HintTech onze eerste nieuwe partner.’

USoft verkoopt het eigen product, maar maakt tevens webgebaseerde oplossingen en applicaties voor marktsegmenten, zoals woningcorporaties, zorgverzekeraars en pensioenverzekeraars. USoft werkt onder andere voor Achmea, Agentschap NL, Menzis, ONVZ en Vestia.

dan gewijzigd of gedropt moet worden. Dit contrasteert met een proces-georiënteerde implementatie, waar je niet zomaar een stap kunt weghalen zonder dat de keten van benodigde bewerkingen wordt onderbroken.

In de USoft-omgeving worden koppelingen tussen specificaties en implementaties handmatig geregistreerd door de teamleden. De grote belofte achter conceptuele modellen zoals SBVR is dat in de toekomst de implementaties automatisch uit de natuurlijke-taal regels ge(her)genereerd zouden kunnen worden. Dan zouden business owners eindelijk het hele bedrijfssysteem kunnen aanpassen zonder tussenkomst van IT specialisten. Zover is het nog lang niet. Voorlopig biedt deze nieuwe methodiek veeleer de mogelijkheid om opdrachtgevers voor hetzelfde budget meer te bieden, vooral op het gebied van aanpasbaarheid.

Deze methodiek biedt echt een andere invalshoek dan het vertrouwde object-georiënteerd modelleren. Dat blijkt uit tal van details. Conceptuele modellen kunnen zo snel worden omgezet in software, en prototypen zijn zo gemakkelijk werkend te krijgen en te vullen met voor de opdrachtgever relevante voorbeelddata, dat er geen poging wordt gedaan om herbruikbare bibliotheken te onderhouden: elke applicatie wordt vanaf de grond opgebouwd. Encapsulatie en het definiëren van interfaces tussen modules, een kenmerk van service-oriented architectuur, speelt helemaal geen centrale rol. De nadruk ligt juist op het flexibel kunnen aanbrengen van onvoorziene afhankelijkheden tussen bedrijfsregels, waardoor eerder een ononderbroken en applicatiebreed web van afhankelijkheden ontstaat dan services met hun vooropgestelde scheiding van verantwoordelijkheden.

Natuurlijk kan er ook in deze omgeving zonder bezwaar sprake zijn van encapsulatie en interfacing, maar pas wanneer de bedrijfslogica dit specificeert, niet a priori omdat het een uitgangspunt is van de ontwikkelmethodiek.

OMG heeft met SBVR dan ook werkelijk een computation-independent standaard afgeleverd, die niet al is ontworpen met een scheef oog op geschiktheid voor hun eigen model-driven architecture (MDA) en UML modellering. Dit USoft project bewijst dat zij hierin geslaagd zijn, een niet geringe prestatie. In de standaard is naast de UML canon duidelijk de invloed merkbaar van alternatieve benaderingen, zoals The Business Rules Group, NIAM (nu FCO-IM) en ORM (Object Role Modelling). Weliswaar specificeert OMG met welke technieken de inpassing van SBVR in MDA kan gebeuren (XMI, MOF) maar deze vertaalslagen staan in optionele, afzonderlijke hoofdstukken.

Business processes versus business rules

Business rules en business processes voorstanders lijken nog altijd op gespannen voet te staan, maar het verband tussen de twee kan op veel manieren praktisch worden vormgegeven. In het hier beschreven project ligt de rules engine als een beschermende laag om alle bedrijfsgegevens heen. Daar horen ook de gegevens bij die door processen met een specifieke workflow worden gevormd of aangesproken. Daarom worden bedrijfsprocessen en alle daarmee verbonden concepten simpelweg als een subset gezien: een categorie business rules. Relevante procesdata, zoals informatie over de bereikte status bij de vorming van een contract, of de routing van documenten door de organisatie, worden “meegemodelleerd” in het datamodel. Er

wordt niet speciaal een poging gedaan om procesmatige aspecten te scheiden van andere aspecten (zoals data-integriteit, software interface ontwerp, gegevensopslag, autorisatie...), wat op zich al een zorg minder is in een agile project.

Maar het is ook goed mogelijk om dit soort rules engine uitsluitend voor data integriteit en -kwaliteit in te zetten, en er een workflow of orchestration engine “bovenop te zetten” voor een gescheiden gehouden laag van proceslogica. Een derde veel toegepaste mogelijkheid is om een rules engine geheel buiten het primaire systeem op te stellen. Het is dan een expert module die alleen als dienst wordt aangeroepen voor beslissingen die veel rekenkracht en aanpasbaarheid vereisen, zoals het doorrekenen van wettelijke voorwaarden en percentages bij belastingheffing of het beantwoorden van de vraag of iemand in aanmerking komt voor een lening of uitkering.

Waarom business rules en agile?

Het USoft-project maakt de natuurlijke affiniteit tussen business rules en agile goed duidelijk:

- Constante communicatie met en bijsturing door de opdrachtgever worden vergemakkelijkt door het gebruik van natuurlijke-taal regels.
- De vorm van elementaire, op zichzelf ware of onware beweringen is voor de klant erg makkelijk om te valideren.
- De in 4GL/RAD gewortelde ontwikkeltools zijn in staat snel prototypen te bieden, dus werkende schermen met “echte” voorbeelddata, ideaal voor overleg met de opdrachtgever.
- De prototypen en business rules worden nooit weggegooid, maar altijd al of niet doorontwikkeld. Business rules en hun implementaties kunnen één voor één snel een hogere of lagere prioriteit krijgen. Dit alles komt goed overeen met het agile uitgangspunt om goed-genoeg software te leveren, ofwel zoveel mogelijk te leveren binnen de beschikbare tijd en budget.
- Aanpasbaarheid, kapitaal voor het slagen van agile, wordt vergroot doordat de meta-data volledig open en raadpleegbaar zijn. Het meta-data model is niet ingekapseld in een pakket of dienst die alleen een interface publiceert.

Wanneer deze aanpak?

Er is geen enkele aanwijzing dat ‘agile 4GL+’ alleen kansrijk zou zijn bij projecten of systemen tot een bepaalde omvang. Wel is een inherent kenmerk van agile dat klein begonnen wordt, met een pilot of een behapbaar stuk functionaliteit.

‘Agile 4GL+’ is gericht op het gestructureerd opslaan van meta-data en applicatie data en daarom vooral geschikt voor bedrijfssoftware die data-intensief is, waarin de meeste eindgebruikershandelingen zullen bestaan uit het opvragen of manipuleren van

gestructureerde gegevens, en waarbij eindgebruikers enige training kunnen krijgen.

Acceptabele performance zal ongetwijfeld een eis zijn, maar snelheid is geen concurrerend kenmerk van ‘agile 4GL+’ oplossingen. De aanpak is dus typisch geschikt voor administratieve toepassingen en minder voor real-time systemen voor verwerking van beurskoersen of alarmsystemen. Door het gebruik van business rules is de methodiek wel bij uitstek geschikt voor databases waarvoor een hoog niveau van data kwaliteit aantoonbaar moet zijn. ‘Agile 4GL+’ is geschikter naarmate meer maatwerk is gewenst en dat maatwerk ook heel snel weer moet kunnen worden aangepast. Voor standaard spreadsheet toepassingen, documentbeheer of boekhoudkundige handelingen zijn pakketten concurrerder.

Dit project maakt gebruik van een rules engine product. Dit soort producten is nog niet wijdverbreid. Meestal moet nieuw- of aanbouw niet alleen zonder rules engine plaatsvinden, maar ook nog in een omgeving die al bepaald wordt door een heel andere methodiek, zoals workflow-gerichte architectuur of SOA, ERP en CRM pakketten, en legacy databases. Ook onder dit soort omstandigheden kan ‘Agile 4GL+’ een inspiratiebron zijn, zeker als er een 4GL tool aanwezig is. Met specificatie van business rules kan al goed begonnen worden door een database tabel in te richten waarin per record een business rule en de daarbij gewenste statusinformatie wordt opgeslagen.

De methodiek is veel minder ingeburgerd dan object-georiënteerd modelleren. Het kan een uitdaging zijn om geschikte teamleden te vinden. Zij moeten goed zijn in informatie analyse, relationeel modelleren en SQL, maar hoeven geen programmeerkennis te hebben.

‘Agile 4GL+’ slaagt alleen als de teamleden het beheer krijgen over de database. Wat elders in code wordt uitgedrukt vertaalt zich hier direct in data structuur. Uiteraard vereisen structuurwijzigingen meer beleid in een productie omgeving dan tijdens ontwikkeling. Maar als het beheer van productiedata exclusief is belegd bij een andere, afzonderlijk opererende afdeling, ontstaat het gevaar van ernstige stagnatie, dat beter vooraf kan worden weggenomen.

Zoals iedereen weet die agile heeft gewerkt, is het allerbelangrijkste dat de opdrachtgever bereid en in staat is om op deze vernieuwende manier mee te werken, waarbij hij enerzijds meer invloed, inspraak en inzicht in eigen gegevens krijgt, maar zich anderzijds ook meer betrokken moet tonen. «

Referenties

- Zie voor een kort overzicht van de business rules aanpak bijvoorbeeld www.businessrules-group.org/brmanifesto.htm. Uitgever: Coenen, Alcedo, Leo Hermans, Matthijs van Roosmalen en Silvie Spreeuwenberg, (Uw bedrijf geregeld met) Business Rule Management, Academic Service, 2008.
- SBVR: een goede inleiding is Annex A van de specificatie zelf. Versie 1.0, januari 2008, is gratis en anoniem te downloaden van www.omg.org/spec/SBVR/1.0/ (4MB PDF).
- Agile: een kort, goed en concreet overzicht dat past bij dit artikel is www.dsdm.org/products/atem.asp (registratie verplicht, online lezen gratis, downloaden niet gratis).
- Het rood-groen-blaauwe kleurcoderen van tekst is een informele aanbeveling van SBVR en valt buiten de eigenlijke standaard. Hier gebruiken we voor het vocabulaire gedeelte, net als voor de bedrijfsregels, beweringen; SBVR gebruikt eerder alfabetische woordenboeklemma's, maar dit is een onbelangrijk detail.
- Chisholm, Malcolm, De connectie tussen business rules en databases, in: Database Magazine, oktober 2008, nr. 6.