Legacy-systemen blijven betrouwbare schakels in modernisering

Ongestructureerde data ontsluiten

Hans Peter van der Horst

Bijna elk groot bedrijf bezit wel een legacy-omgeving; een omgeving waarin grote hoeveelheden records real-time worden verwerkt. Daarnaast is modernisering van het applicatielandschap aan de orde. Dat roept uiteraard een aantal vraagstukken op, onder andere hoe ongestructureerde data real-time worden verwerkt.

Een dergelijk probleem speelde bij een grote opdrachtgever op financieel terrein. In dit specifieke geval gaat het om legacy-systemen die op het HP OpenVMS platform operationeel zijn en geschreven zijn in traditionele talen zoals Cobol en C. Deze systemen moeten aansluiten op een technologisch modern Operational Data Store (ODS) systeem dat gegevens beschikbaar stelt in XML formaat. De legacy-systemen kunnen opdrachten geven via een message queuing systeem aan het ODS systeem. Ontsluiten van de informatie is geen probleem, dat kan via het message queuing systeem.
Het vraagstuk is hoe er voor wordt gezorgd dat het OpenVMS Cobol/C systeem XML genereert. Indien dit niet lukt, moet er een alternatief worden gezocht, waarbij de volgende uitgangspunten gehanteerd worden: een schaalbare oplossing; grote volumes aankunnen; 24x7 beschikbaarheid; real-time afhandeling met uitstekende responstijden (< 0,2 seconden); bewezen software; door derde partij ondersteunde sofware.

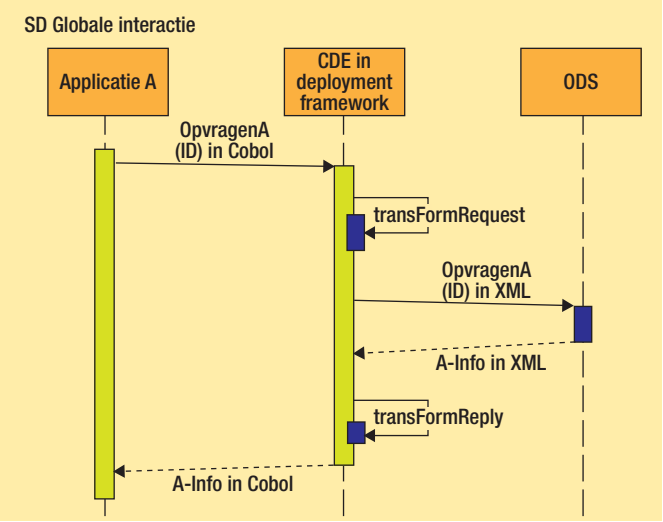
Alternatieven

Het ligt voor de hand om het OpenVMS platform als uitgangspunt te nemen omdat daarop de applicaties operationeel zijn. Onderzoek leerde dat het adopteren van recente XML standaarden hierbij een knelpunt vormde. De bibliotheken die nodig zijn om XML te genereren blijken de nieuwe standaarden niet te volgen. Het aanpakken van *porting* werk om er voor te zorgen dat met behulp van de Apache APR bibliotheek toch beschikt kan worden over de nieuwe standaarden, lag niet op het pad. De uitgangspunten 'bewezen' en 'ondersteunde software' worden hiermee gepasseerd. Hierdoor ontstond de noodzaak om een verkenning te doen met betrekking tot welke alternatieven voorhanden zijn.
Zo blijkt er een oplossing te zijn die in staat is om vanuit een Cobol linkrecord een XML bericht te genereren en weer terug. Er wordt Cobol programmatuur gegenereerd die netjes alle tags maakt om de opdracht in XML om te zetten. Het antwoord op deze opdracht komt in XML terug en wordt door de gegenereerde Cobol programmatuur ontdaan van alle XML tags en de data worden netjes terug aan de Cobol programmatuur geleverd. Een *Proof of Concept* leerde dat dit in exact één geval functioneerde: het testgeval dat als basis fungeerde voor de generatie van het vraag- en antwoordbericht. Het stellen van een vraag met een ander testgeval gaat overigens prima, het ontvangen van een antwoord gaat hopeloos mis.

> CDE kan met behulp van de Cobol library een Cobol copybook inlezen

De oorzaak ligt in het feit dat de ontvangen data een andere lengte hebben en daarmee een andere opmaak kennen. De Cobol programmatuur raakt het spoor bijster en bij terugkerende data die groter zijn dan in het eerste testgeval, leidt dit tot corruptie van het geheugen: de applicatie crasht. Het element in XML dat problemen veroorzaakt is een *namespace*. Dit vrij te definiëren element zorgt voor de nodige variatie in grootte van het XML bericht, waar de Cobol programmatuur niet tegen kan. Dit bracht ons weer terug bij de tekentafel. Een eenvoudige, adequate oplossing bleek niet direct beschikbaar te zijn.

Nader onderzoek leerde dat het verstandig is om naar alternatieve hardware platforms te kijken en daarmee alternatieve producten toegankelijk te maken. Het aanschakelen van alternatieve hardware platforms is binnen een message queuing infrastructuur relatief eenvoudig. Het vereist in deze case een



Afbeelding 1: UML Sequencediagram die op globale wijze de interactie weergeeft.

aanpassing in de routering waardoor er een extra schakel in de keten wordt geïntroduceerd. Deze schakel neemt dan de conversie van Cobol Linkrecord naar XML bericht voor haar rekening.

CDE

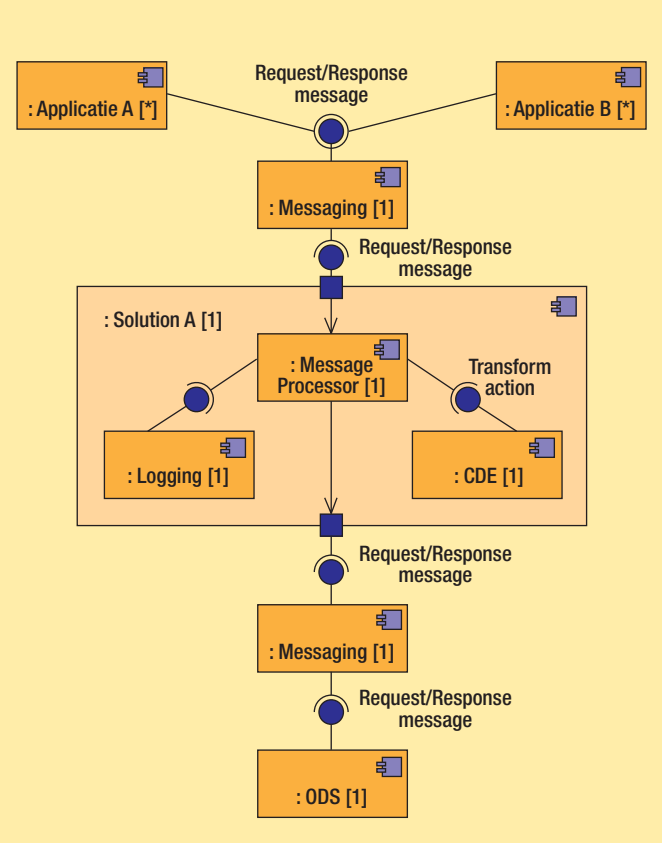
Alhoewel er talrijke oplossingen op de markt voorhanden zijn die in staat zijn datatransformaties uit te voeren, zijn er niet veel die enkel en alleen datatransformaties uitvoeren. Veel oplossingen kennen verplichte 'unattended workflow' voorzieningen, waarbij de oplossing elke ondernomen proces(sing)stap zorgvuldig orkestreert. Het orkestreren voert meestal de boventoon van de functionaliteit van deze oplossingen, waarbij de datatransformatie een nevenfunctie is. Het orkestreren zorgt vaak voor een zeer significante performance degradatie, maar levert natuurlijk een grotere mate van flexibiliteit op. Het is dan ook kwestie van evaluatie of dit voordeel van flexibiliteit opweegt tegen het performance nadeel. In deze case is dit niet het geval. Opnieuw is een Proof of Concept uitgevoerd op basis van drie oplossingen:

- een reeds beschikbare maatwerkoplossing op basis van .NET;
- een reeds in een andere case operationele standaardoplossing met verplichte 'unattended workflow';
- een oplossing op basis van Informatica's Complex Data Exchange, uitgebreid met een deployment framework van Getronics PinkRoccade.

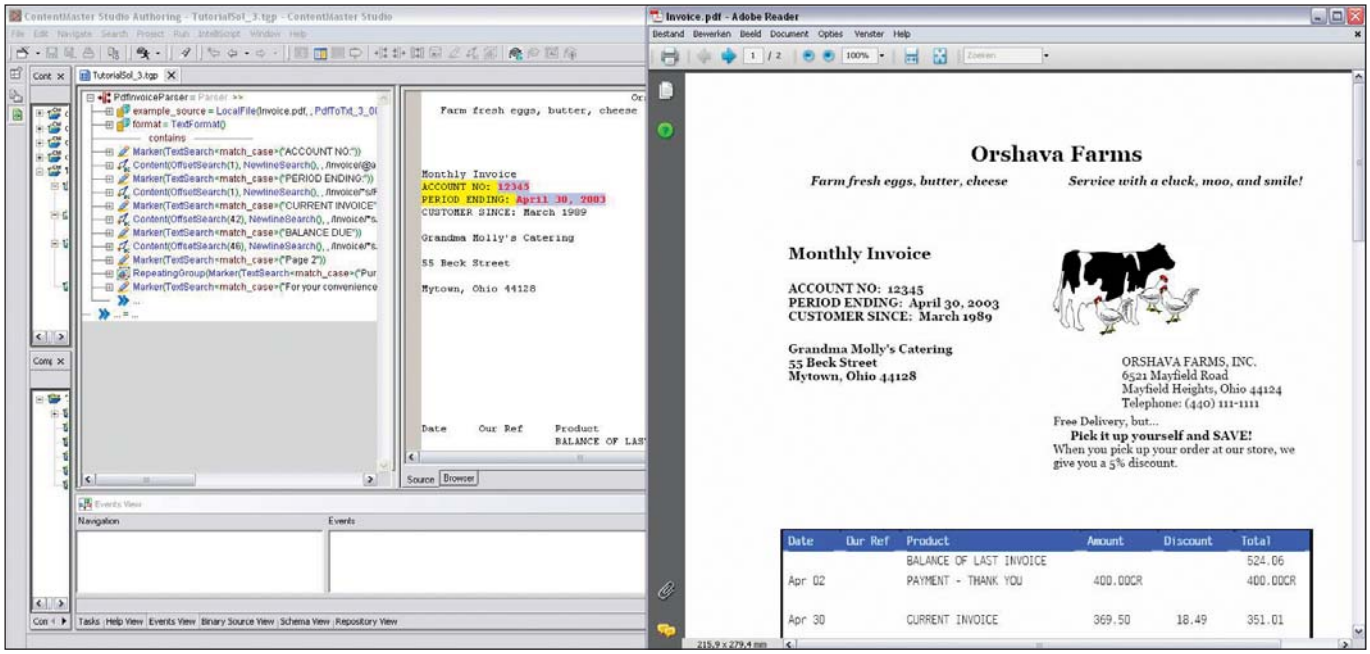
In eerste instantie is gebleken dat alle drie de oplossingen functioneel gezien in staat zijn de Cobol linkrecord vraagopdrachten aan te nemen, te transformeren en een XML vraagopdracht door te sturen naar ODS, zie afbeelding 1. Voorts lukt het ook om de verschillende antwoordopdrachten in XML volledig terug te transformeren naar een Cobol linkrecord antwoordbericht.
Een benchmark tussen deze drie oplossingen leverde het volgende op:

- Oplossing 1 kende een 25 procent slechtere performance ten opzichte van oplossing 2;
- Oplossing 3 kende een 200 procent betere performance ten opzichte van oplossing 2;
- Oplossing 1 kende in zijn implementatie een schaalbaarheid tot twee threads;
- Oplossing 2 kende zijn optimum in acht threads;
- Oplossing 3 schaalde lineair door, er is gestopt bij tien threads.

Informatica's Complex Data Exchange (voorheen ContentMaster, in december 2006 verkregen door overname van Itemfield door Informatica), kortweg CDE, is in staat ongestructureerde data razendsnel te transformeren naar gestructureerde data. Het is een oplossing die in staat is conversies te doen van het ene willekeurige formaat naar het andere willekeurige formaat.
De case die besproken wordt zal naast de huidige ook geschikt moeten zijn om toekomstige (functionele) dialecten te transformeren naar diverse (XML) taxonomieën. Het Informatica product blijkt hier geschikt voor te zijn en in staat te zijn door middel van taxonomiebibliotheken (zoals HL7, EDIFact, SWIFT, SEPA, ACORD enzovoort) bij te blijven met de actuele stand van zaken met betrekking tot de (XML) standaarden. De gerealiseerde transformatieregels zijn opgeslagen in een formaat dat naar alle platforms te plaatsen is waar een runtime engine beschikbaar is van CDE. Deze engine pakt deze regels op en kan dan runtime de conversies uitvoeren.
Informatica heeft voor en samen met diverse technologieleveranciers agents gemaakt zodat deze krachtige transformatie-



Afbeelding 2: UML componentdiagram voor de beschreven case.



Afbeelding 3: De studio met het herkennen en markeren van een PDF document.

tool direct in de ontwikkelomgevingen opgenomen kan worden. Zo zijn er agents voor SAP Netweaver, IBM WebSphere Business Integration Message Broker, Microsoft BizTalk, webMethods en Oracle BPEL.

Deployment framework

Om op de bestaande message queuing in de beschreven case aan te sluiten, is er voor gekozen om de CDE-API te gebruiken om een eigen agent te realiseren. Deze is zelfs zo gebouwd dat op basis van de in een andere specifieke case geldende eisen, eenvoudig aangesloten kan worden op een willekeurige andere infrastructuur. Denk hierbij aan FTP, JMS, Webservice en databaseverbindingen. Hiermee is getracht om deze transformatie oplossing ontsluitbaar te maken in een veelvoud van situaties via een deployment framework. De kern, namelijk de vertaalslag zelf, is op basis van transformatieregels in de ontwikkelomgeving (studio) van CDE gerealiseerd. Noodzakelijke voorzieningen zoals logging en tracing zijn eveneens opgenomen in dit deployment framework.

De ontwikkelomgeving is op Eclipse gebaseerd en voorziet in het importeren van datastructuren. Deze datastructuren worden op basis van geïnstalleerde bibliotheken herkend. Met enige kennis kunnen zelf ook bibliotheken worden gemaakt. Hiervoor geldt geen exclusiviteit vanuit Informatica, hulp kan daarbij worden geboden. De datastructuur waarnaar de data getransformeerd worden, wordt op identieke wijze geselecteerd via een bibliotheek. CDE kan met behulp van de Cobol library een Cobol copybook inlezen en daarmee een linkrecord tot zich nemen. Op basis van dit linkrecord wordt er een XSD (XML Schema Definition) gegenereerd dat een afspiegeling is van het Cobol Linkrecord. Vervolgens is er een parser die de input data conform het Cobol linkrecord transformeert naar XML. In de Eclipse gebaseerde ontwikkelomgeving (studio) van CDE worden er mappings gemaakt van hoe input data-elementen passen op output data-elementen. Tenslotte is er een serializer waarin de transformatie van XML naar Cobol wordt uitgevoerd.

PDF

CDE is ook in staat om data-elementen van een PDF document te verkrijgen, zie afbeelding 3. PDF documenten zoals facturen, opdrachtbevestigingen of bestelopdrachten kunnen met behulp van de juiste bibliotheken in de studio worden geïmporteerd. Vervolgens kan dit document als sjabloon worden gebruikt en data-elementen worden gemarkeerd. Hierdoor wordt het mogelijk om op basis van een PDF document een systeem te voorzien van data die in deze documenten zijn opgeslagen. Leveranciergebonden formaten zoals Word, Excel, maar ook SGML, Postscript en RTF kunnen worden geïmporteerd. Verder geautomatiseerde verwerking komt hiermee binnen handbereik. Zo worden ongestructureerde data toegankelijk als belangrijke invoer, voor legacy-systemen, maar ook voor web-systemen waarin een geconsolideerd beeld van de beschikbare informatie moet worden getoond. De legacy-systemen, vaak de hoekstenen van de informatievoorziening, zijn daarmee betrouwbare schakels in de modernisering van het applicatielandschap. Deze oplossing biedt de mogelijkheid om meer fasering toe te passen en een risicovolle Big-Bang uit de weg te gaan. Meer informatie over Complex Data Exchange is te vinden op www.informatica.com/solutions/complex_data/default.htm.

Hans Peter van der Horst

Ing. H. P. van der Horst (hanspeter.vanderhorst@getronics.com) is Solution & Product Manager Enterprise Integration Solutions bij Getronics PinkRoccade. Met dank aan drs. Martin Haubrich.