

Ontwikkeltools voor softwarecomponenten - de stand van zaken

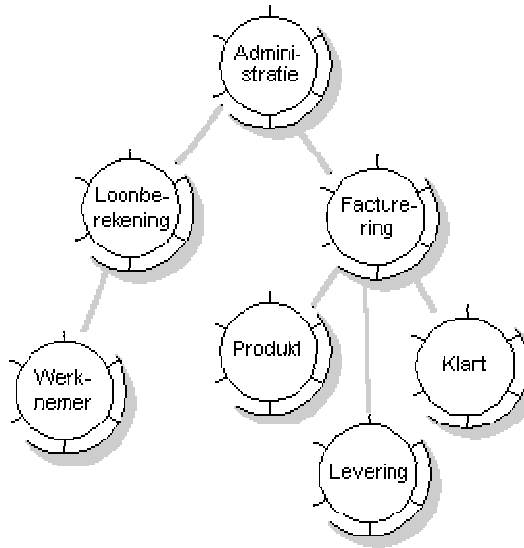
Huidige generatie ontwikkeltools geschikt voor het snel opleveren van schaalbare two-tier, three-tier en multi-tier client/server applicaties.

Vandaag de dag hebben organisaties nog altijd te kampen met een groot aantal problemen bij het gebruik van informatietechnologie: de ontwikkel- en onderhoudskosten van de informatievoorziening van een organisatie zijn hoog, het beschikbaar stellen van de juiste gegevens op het juiste tijdstip en de juiste plaats vergt een grote inspanning, de bestaande informatiesystemen belemmeren het snel invoeren van nieuwe producten en diensten, het in de lucht houden van de informatiesystemen kost veel geld en het aanpassen van de informatiesystemen aan de veranderende behoefte kost nog meer geld.

Het zijn problemen die al een geruime tijd bestaan en waarvoor de software-industrie tot voor kort geen concrete oplossing had. Tot voor kort, want één van de ideeën om deze *software crisis* op te lossen lijkt navolging te krijgen.

Softwarecomponenten

Het concept voor een goede informatievoorziening is *ontkoppeling*. Ontkoppeling houdt in dat de informatievoorziening wordt gerealiseerd middels een stelsel van onafhankelijke softwarecomponenten die via een infrastructuur (middleware) met elkaar communiceren.



Figuur 1 Onafhankelijk software componenten die onderling communiceren.

Zo wordt de informatiebehoefte van de afdeling *administratie*, die tot taak heeft het berekenen van het loon van een werknemer en het factureren van de geleverde producten aan een klant, gerealiseerd door een aantal onafhankelijk softwarecomponenten: *Administratie*, *Loonberekening*, *Facturering*, *Werknemer*, *Produkt*, *Levering* en *Klant* die onderling communiceren (zie figuur één).

Iedere softwarecomponent biedt middels een *interface* een aantal *diensten* aan. Communicatie tussen componenten verloopt altijd via de interfaces. Wanneer een component gebruik maakt van de diensten van een andere component, wordt er tussen de componenten een *contract* afgesloten. Dit contract bevat afspraken over de beschikbaarheid van de dienst, de gegarandeerde responsietijd en andere operationele afspraken.

Softwarecomponenten gaan verder dan de bekende *softwaremodules*. Bij softwaremodules blijft de ontkoppeling beperkt tot het programma of systeem waartoe de modules behoren. Bij softwarecomponenten gaat het om ontkoppeling op een hoger niveau. Systemen en programma's bestaan niet meer. Iedere softwarecomponent neemt een specifiek deel van de totale functionaliteit voor zijn rekening. Applicaties ontstaan doordat componenten op een bepaalde manier met elkaar samenwerken.

Het realiseren van de informatievoorziening door middel van een stelsel onafhankelijke softwarecomponenten beperkt de effecten en daarmee ook de kosten van verandering. Doordat applicatielogica en gegevens met andere componenten gedeeld kunnen worden, doen zich minder problemen voor met redundantie en kunnen gegevens op het juiste tijdstip op de juiste plaats beschikbaar gesteld worden.

Evaluatieproject

Tegenwoordig kan een softwareontwikkelaar kiezen uit een aantal ontwikkeltools om softwarecomponenten die middels een infrastructuur met elkaar communiceren, te ontwikkelen. Voor SERC en Computable de rede om een evaluatieproject te organiseren om zo de stand van zaken te inventariseren.

Dit evaluatieproject was het zesde in de serie evaluatieprojecten en had tot doel om teams van ontwikkelaars kennis te laten maken met ontwikkeltools voor softwarecomponenten. De vijf voorafgaande evaluatieprojecten hadden betrekking op object oriëntatie, grafische gebruikersinterfaces, workflow management en groupware. Naar alle waarschijnlijkheid zal het volgende evaluatieproject in het teken staan van Case-tools voor object georiënteerd modelleren.

De volgende ontwikkeltools werden geëvalueerd:

- Composer geleverd door Texas Instruments Software BV
- Dynasty geleverd door Value Added Consultancy BV
- NatStar geleverd door NatSystems Benelux NV
- ObjectStar geleverd door Amdahl Nederland BV
- Prolifics geleverd door Brink Automatisering BV
- SuperNova geleverd door SuperNova International BV
- Uniface geleverd door Compuware BV
- USoft geleverd door USoft Benelux BV
- VisualAge Generator geleverd door IBM Nederland NV

Uiteraard was het niet mogelijk om alle beschikbare ontwikkeltools te evalueren, maar is op basis van: het marktaandeel van het ontwikkeltool, de beschikbaarheid van het ontwikkeltool in Nederland en bereidwilligheid van de leverancier om mee te werken aan evaluatieproject, bovenstaande selectie gemaakt.

Het evaluatieproject was opgedeeld in drie stappen:

Leren - In deze stap werden de deelnemers van het evaluatieproject bekend gemaakt met de theorie van softwarecomponenten en de verschillende visies op ont koppeling door Ir. G.M.A. van der Harst en Ing. G.F. Florijn. Tevens werd er een korte inleiding gegeven door de deelnemende ontwikkeltool leveranciers.

Experimenteren - Door het oplossen van een door SERC geleverde casus werd vervolgens praktisch geëxperimenteerd met de verschillende ontwikkeltools. De casus bestond uit het realiseren van een prototype van een flexibel factureringssysteem voor het telecommunicatiebedrijf Flextel op basis van een gegeven object model. De volgende bedrijfsprocessen moesten door het prototype ondersteund worden: het aanmelden van een nieuwe klant, het wijzigingen van prijsafspraken met bestaande klanten, het genereren van marketing overzichten en de facturering. Na implementatie van de opdracht moesten een drietal veranderingen aangebracht worden: het toevoegen van een nieuwe prijsregeling, het vastleggen van factureringafspraken en het opsporen van zogenaamde vreemde telefoontjes.

Evalueren - Ter afsluiting presenteerde elk ontwikkelteam zijn bevindingen omtrent het realiseren van de casus met het gebruikte ontwikkeltool.

Op basis van de bevindingen wordt de stand van zaken met betrekking tot softwarecomponenten geïnventariseerd. De rest van het artikel gaat in op deze bevindingen. Eerst zal stilgestaan worden bij de visie op ont koppeling die door de verschillende ontwikkeltools gehanteerd wordt. Vervolgens komen de ontwikkel- en beheersaspecten van de verschillende ontwikkeltools aan bod. Voor een gedetailleerde uitwerking van de resultaten wordt verwezen naar het eindrapport van het evaluatieproject [2].

Visie op ont koppeling

Hoewel alle tools uitgaan van het ontwikkelen van softwarecomponenten is de wijze waarop deze logisch worden ingedeeld (de ont koppelingsvisie) verschillend. De tools kunnen ingedeeld worden in de volgende categorieën:

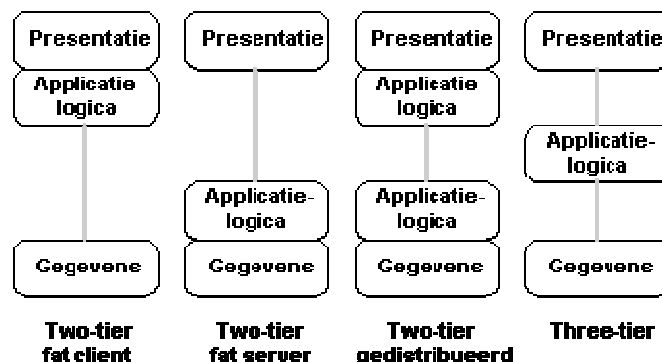
Two-tier client/server

Onder deze categorie vallen de tools ObjectStar en USoft. Beide tools onderscheiden twee typen softwarecomponenten: clients en servers. Grofweg zijn er drie varianten te onderscheiden: de *fat client* variant waarbij de applicatielogica op de clients draait, de *fat server* variant waarbij de applicatielogica in de server is ondergebracht en een *gedistribueerde* variant waarbij applicatielogica zowel in de clients als in de server zijn ondergebracht (zie figuur twee). Doordat bij de *fat client* variant alle applicatielogica op de client executeert, vindt er duplicatie van applicatielogica plaats. Dit brengt distributie- en consistentieproblemen met zich mee. Een ander probleem is de belasting van het netwerk door de grote hoeveelheid gegevens die van de server naar de clients getransporteerd moet worden. Het probleem van de *fat server* variant is dat de performance van de applicatie negatief beïnvloed wordt bij een toename van het aantal clients. De gedistribueerde variant lost de performance problemen enigszins op maar maakt de oplossing door de verdeling van applicatielogica minder inzichtelijk.

Hoewel door Amdahl (leverancier van ObjectStar) het gebruik van de *fat server* variant geprefereerd wordt, zijn alle two-tier client/server varianten in principe mogelijk met ObjectStar. Bij USoft wordt de *fat client* variant toegepast waarbij presentatielogica en bedrijfsregels op de client executeren.

Three-tier client/server

De ontwikkeltools waarbij softwarecomponenten in drie typen worden onderverdeeld zijn: Dynasty, NatStar, Prolifics en Uniface. Bij elk van deze ontwikkeltools gaat het om een presentatielaag, een service laag en een gegevenslaag. De presentatielaag bevat de presentatielogica componenten, de service laag de applicatielogica componenten en de gegevenslaag de database server waarin de gegevens worden vastgelegd (zie figuur twee).



Figuur 2 Van de two-tier client/server varianten naar three-tier client/server.

Dynasty onderscheidt zich van de drie andere tools doordat op de service laag objecten gedefinieerd worden. Hiermee wordt de ontkoppelvingsvisie van gedistribueerde objecten benaderd (zie gedistribueerde objecten).

Multi-tier client/server

Als derde categorie worden ontwikkeltools onderscheiden die een minder "strakke" typering van softwarecomponenten voorschrijven. Alleen de presentatielaag en de gegevenslaag zijn vast getypeerd, de lagen ertussen zijn niet getypeerd. Hiermee is het aantal lagen waarin de softwarecomponenten kunnen worden ingedeeld in principe oneindig. Ontwikkeltools die in deze categorie vallen zijn: Composer, SuperNova en VisualAge Generator.

Gedistribueerde objecten

De laatste categorie gaat nóg een stap verder dan de multi-tier client/server benadering. Het opdelen van een informatiesysteem in softwarecomponenten geschiedt op basis van de ideeën van *object-oriëntatie* (OO). Gegevens en functionaliteit die in de andere benaderingen nog van elkaar gescheiden zijn, komen samen in een object. Gedistribueerde objecten kunnen samenwerken met andere objecten buiten het eigen programma. Een gedistribueerd object kan zelfs samenwerken met objecten op andere machines ongeacht besturingssysteem, netwerk, programmeertaal en hardware. Dit in tegenstelling tot "gewone" objecten van een programma. Deze kunnen alleen gebruik maken van de diensten van andere objecten binnen het programma.

Daar door de deelnemers van het evaluatieproject duidelijk naar voren werd gebracht dat er behoefte is aan het ontwikkelen van een bedrijfsbrede informatievoorziening middels onafhankelijke, gedistribueerde objecten, is het opvallend dat geen van de geëvalueerde ontwikkeltools onder deze categorie valt.

Bij bovenstaande opdeling moet de volgende kanttekening worden gemaakt: voor three-tier client/server tools geldt dat ook two-tier client/server applicatie ontwikkeld kunnen worden; voor multi-tier client/server tools geldt dat naast multi-tier client/server applicaties ook three-tier client/server applicaties en two-tier client/server applicaties opgeleverd kunnen worden. In tabel één wordt een overzicht gegeven van de visie op ont koppeling door de verschillende ontwikkeltools.

Tool	Two-tier	Three-tier	Multi-tier
Composer	x	x	x
Dynasty	x	x	
NatStar	x	x	
ObjectStar	x		
Prolifics	x	x	
SuperNova	x	x	x
Uniface	x	x	

USoft	x		
VisualAge Generator	x	x	x

Tabel 1 Visie op ont koppeling.

Ontwikkelaspecten

Wanneer de ontwikkeltools bekeken worden naar de ondersteunde fasen van het *ontwikkelproces* kunnen drie categorieën onderscheiden worden:

Analyse, ontwerp en implementatie

Sommige tools richten zich op de ondersteuning van het volledige ontwikkelproces: Composer en NatStar. Bij deze tools is het in principe niet nodig om een apart CASE-tool te gebruiken. Voordeel van deze ondersteuning is dat alle modellen met één tool kunnen worden gemaakt en deze op één plek (repository) vastgelegd worden. Daarnaast is het met beide tools mogelijk om op basis van de modellen de implementatie te genereren.

Ontwerp en implementatie

Met ontwikkeltools als Dynasty en USoft is het mogelijk een deel van het ontwerp met het ontwikkeltool te doen. Beide tools volgen de filosofie dat eerst met een aparte CASE-tool een gegevensmodel wordt opgebouwd. Op basis van dit gegevensmodel kunnen vervolgens in Dynasty objecten en met USoft *bedrijfsregels* gemodelleerd worden.

Implementatie

De overige ontwikkeltools (ObjectStar, Prolifics, SuperNova, Uniface en VisualAge Generator) richten zich op de implementatiefase. Deze gaan er vanuit dat de analyse- en ontwerp-fase reeds zijn uitgevoerd.

In tabel twee wordt een overzicht van de ondersteunde fasen van het ontwikkelproces door de verschillende ontwikkeltools.

Tool	Implementatie	Ontwerp	Analyse
Composer	x	x	x
Dynasty	x	x	
NatStar	x	x	x
ObjectStar	x		
Prolifics	x		
SuperNova	x		
Uniface	x		
USoft	x	x	
VisualAge Generator	x		

Tabel 2 Visie op ont koppeling.

Het genereren van softwarecomponenten kan geschieden op basis van gegevensdefinities (bijvoorbeeld in de vorm van gegevensmodellen of object modellen) of op basis van gespecificeerde logica (bijvoorbeeld in de vorm van actiediagrammen of bedrijfslogica). De meeste tools genereren op basis van gegevensdefinities: Dynasty, Prolifics, Uniface en VisualAge Generator. De tools die genereren op basis van beide zijn Composer, NatStar en USoft. Bij ObjectStar en SuperNova is generatie beperkt mogelijk.

Met uitzondering van ObjectStar en USoft bieden alle ontwikkeltools mogelijkheden voor het definiëren van templates. Deze templates bevorderen *hergebruik* van modellen en code over projecten heen.

Opvallend is dat elke tool zijn eigen programmeertaal en *infrastructuur* gebruikt. Softwarecomponenten kunnen hierdoor niet direct worden overgezet naar een ander ontwikkeltool en communicatie tussen deze componenten is niet zonder meer mogelijk. Hierdoor wordt een afhankelijkheid van de leverancier gecreëerd. USoft en ObjectStar maken voor de interne communicatie gebruik van CORBA. Composer, Dynasty, SuperNova en VisualAge generator bieden wel de mogelijkheid om gebruik te maken van ten minste één van de infrastructuurstandaards (zie tabel 3) voor de communicatie tussen eigen componenten en vreemde componenten. De communicatie tussen eigen componenten onderling vindt dan plaats door middel van de eigen infrastructuur.

Tool	Proprietary	DCE	CORBA	DCOM
Composer	x	x	x	x

Dynasty	x	x	x	
NatStar	x			
ObjectStar	x			
Prolifics	x			
SuperNova	x	x	x	
Uniface	x			
USoft	x			
VisualAge Generator	x	x		

Tabel 3 Ondersteuning van infrastructuurstandaards.

Prolifics en Uniface hebben CORBA en DCOM ondersteuning voor de toekomst aangekondigd.

Beheersaspecten

De *partitioning* van de softwarecomponenten over de beschikbare machines in de run-time omgeving wordt op twee manieren opgelost. Bij ontwikkeltools als Composer, NatStar en USoft moeten softwarecomponenten voor een andere partitionering opnieuw gegenereerd of gecompileerd worden. Bij de andere tools: Dynasty, ObjectStar, Prolifics, SuperNova, Uniface en VisualAge Generator kunnen de softwarecomponenten run-time van de ene naar de andere machine overgezet worden.

De *distributie* van softwarecomponenten over de run-time omgeving wordt door de ontwikkeltool: ObjectStar, Uniface en VisualAge Generator ondersteund. Voor de overige tools geldt dat de distributie met de hand moet worden gedaan. In praktijk betekent dit gebruik maken van speciale distributie software, FTP-faciliteiten of diskettes.

Casus

Bij de realisatie van het prototype van het factureringssysteem voor Flexstel bleek dat de ontwikkeltools Composer en NatStar het gegeven object model direct ingevoerd kon worden doordat beide tools object modelering ondersteunen. Bij andere tools was "plat slaan" van het object model tot een relationeel datamodel nodig. Dit "platslaan" bleek de flexibiliteit van het prototype niet ten goede te komen. Het doorvoeren van de veranderingen in het prototype bleken niet eenvoudig.

Conclusies

Concluderend kunnen we stellen dat de huidige generatie ontwikkeltools geschikt is voor het *snel* opleveren van *schaalbare* two-tier, three-tier en multi-tier client/server applicaties. De schaalbaarheid wordt verkregen door een service laag en/of transactiemonitor te introduceren tussen de clients en de database servers. Hierdoor zijn grotere aantallen transacties mogelijk en kan het aantal clients groeien. De snelheid van het ontwikkelen wordt met name verkregen door de uitgebreide generatiemogelijkheden. Hierdoor kan in relatief korte tijd een complexe gedistribueerde applicatie worden opgeleverd.

De ontwikkeltools zijn momenteel echter minder geschikt voor het ontwikkelen en beheren van bedrijfsbrede two-tier, three-tier en multi-tier client/server oplossingen. Dit omdat de tools niet uit gaan van het opleveren van onafhankelijke softwarecomponenten maar van applicaties. De softwarecomponenten worden met behulp van een proprietary programmeertaal ontwikkeld en communiceren middels een proprietary infrastructuur. Weliswaar zijn er ontwikkeltools die de mogelijkheid bieden om middels infrastructuurstandaarden (CORBA, DCE of DCOM) te communiceren met andere componenten maar de communicatie tussen "eigen" componenten blijft middels de proprietary infrastructuur plaatsvinden.

De ontwikkeling van softwarecomponenten op basis van de ideeën achter gedistribueerde objecten wordt door de geëvalueerde ontwikkeltools op dit moment nog niet ondersteund.

Auteurs

- Drs. P.J. Koning is Researcher/Consultant bij het Software Engineering Research Centre.
- Ir. G.M.A. van de Harst is Project Manager bij het Software Engineering Research Centre.

Literatuur

- [1] P.J. Koning, G.M.A. van der Harst en G.F. Florijn, 'Multitier'-architectuur biedt flexibiliteit - Onderhoudbare informatievoorziening met behulp van softwarecomponenten, Computable, 7 februari 1997.
 [2] G.M.A. van der Harst, Multi-tier client/server evaluatieproject - Eindrapport, SERC, juni 1997.

Multi-tier client/server evaluatieproject eindrapport

Van het multi-tier client/server evaluatieproject dat SERC in samenwerking met Computable van zeven tot en met elf april 1997 organiseerde, is een gedetailleerd eindrapport [2] verschenen. Het honderd één bladzijden tellend rapport gaat allereerst in op de achtergrond van softwarecomponenten en behandelt vervolgens per ontwikkeltool de onderdelen,

ontwikkelaspecten, beheersaspecten, gebruikaspecten en koppelingsaspecten. Daarna wordt ingegaan op het ontwikkelen van softwarecomponenten en de casus die de ontwikkelteams hebben uitgewerkt. Ter afsluiting wordt de stand van zaken in kaart gebracht. Het eindrapport is voor 85,- te bestellen bij:

*Software Engineering Research Centre
Postbus 424
3500 AK Utrecht
Telefoon: 030 254 5412
Telefax: 030 254 5948
E-mail: info@serc.nl*