

Op 18 maart 2003 kondigt Microsoft het Dynamic Systems Initiative (DSI) aan. Dit ambitieuze initiatief betreft een architectuur die hardware en software bindt en de processen rondom de IT-lifecycle vereenvoudigt en automatiseert zonder hierbij aan flexibiliteit in te boeten. Microsoft beschouwt DSI als een langetermijndoelstelling waarvoor draagvlak binnen de industrie noodzakelijk is. Inmiddels zijn we meer dan drie jaar verder en is het tijd naar de huidige stand van zaken te kijken.

achtergrond

Microsoft DSI

Beheer voor de IT-lifecycle

De laatste jaren neemt de complexiteit van applicatie-architecturen sterk toe. De klassieke client-server architectuur zoals we die kennen evolueert naar gedistribueerde multi-tier architecturen en webservices. Daarnaast zien we steeds meer dat de hardware en ondersteunende technologieën als een commodity gezien worden. De prijzen van hardware dalen en industriebrede standaardisatie neemt toe. Voor deze situatie waarin de architectuur van software verweven wordt met de onderliggende infrastructuur componenten ziet Microsoft de behoefte aan een functioneel en technisch ondersteunend raamwerk: DSI ofwel Dynamic System Initiative.

AANLEIDING TOT DSI Alvorens deze vraag goed te kunnen beantwoorden moeten we kijken naar de kern van een IT Lifecycle. De essentie van een IT Lifecycle en met name de Application Life Cycle is het feit dat de verschillende fases, in de levenscyclus van een applicatie, integraal worden benaderd vanuit verschillende perspectieven, zoals proces, gereedschap, methodes en communicatie.

Software-ontwikkelprojecten zijn vaak complex en kennisintensief, uitgevoerd door verschillende disciplines, waar sprake is van veel handmatig werk en communicatie. Het is dan ook niet verwonderlijk dat deze projecten niet altijd zo lopen als men verwacht. Door de toenemende complexiteit van applicatiearchitectuur (van monolithische naar gedistribueerde) en infrastructuurarchitectuur zijn de uitdagingen en kosten alleen maar groter geworden.

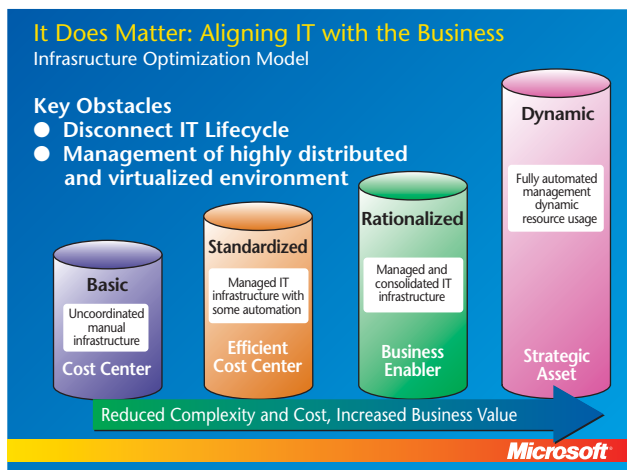
Een andere constatering is dat het merendeel van de kosten gedurende de levenscyclus van een applicatie de

onderhoudskosten zijn (70 tot 80 procent). Een deel van die kosten zijn te verminderen door goede documentatie van de applicatie. De documentatie betreft dan niet alleen het doel en werking van een applicatie maar ook welke eisen deze stelt aan zijn productieomgeving. Een integrale benadering van de levenscyclus van applicatie kan een deel van de bovengenoemde problemen verkleinen door:

- Kennis over het te bouwen systeem en het gehanteerde ontwikkelproces op een eenduidige wijze aan te bieden aan alle betrokken projectleden.
- Communicatie tussen projectleden onderling en met de opdrachtgever te verbeteren.
- Integratie van het gereedschap van de diverse disciplines die betrokken zijn (ook het gereedschap van de beheerder en projectmanager).
- Door ondersteuning van één of meerdere systeemontwikkelmethoden. Het 'Agile Development Process' is de meest genoemde in dit verband.

Het andere deel is de integratie van een development proces en het daaruitvoorkomende product (de applicatie) met de reeds bestaande infrastructuur. Dit vormt vaak een moeizaam proces waarmee hoge kosten gepaard gaan. Omdat dit proces vaak niet goed is ingericht spreekt men van een 'disconnected IT lifecycle'. Hoe beter de development en infrastructuur-omgevingen en processen op elkaar aansluiten hoe effectiever en kostenbesparender de totale oplossing zal zijn.

Microsoft heeft vier 'stadia' gedefinieerd waarin een infrastructuur zich kan bevinden, dit loopt van 'Basic' waarbij de maintenancekosten zeer hoog liggen tot



AFBEELDING 1. De 4 stadia waarin een infrastructuur zich kan bevinden

'Dynamic' waarbij de IT-processen zeer nauw en gedegen op elkaar aansluiten en dus het meest kosteneffectief is (deze laatste is echter zeer moeilijk haalbaar).

MICROSOFT'S ANTWOORD: DSI Microsoft's visie op IT Lifecycle management heet Dynamic Systems Initiative (DSI). Het bijzondere aan deze visie is dat het IT Life Cycle Management verbindt met datacenter automation, het automatiseren van beheeractiviteiten. Een definitie van DSI die overeenkomt met de traditionele benadering van het Application Life Cycle is deze:

'From a core technology perspective, the Dynamic Systems Initiative (DSI) is about building software that enables knowledge of an IT system to be created, modi-

DSI verbindt IT Life Cycle Management met datacenter automation, het automatiseren van beheeractiviteiten

fied, transferred, and operated on throughout the life cycle of that system. These core principles—knowledge, models, and life cycle—are the keys in addressing the complexity and manageability challenges that IT organizations face today.'

Microsoft denkt haar visie op IT Lifecycle management als volgt te kunnen realiseren:

- Door Visual Studio Team System te positioneren als een 'enterprise development' omgeving.
- Door gebruik te maken van modelgedreven ontwikkelgereedschap en geen gebruik te maken van UML-standaards en MDA (Model Driven Architecture).

Microsoft ontwikkelt een eigen standaard, namelijk Domain Specific Languages (DSL). Microsoft meent dat MDA tekort schiet, omdat onder meer MDA zich richt op het applicatieve aspecten en niet de infrastructuurle aspecten van een applicatie.

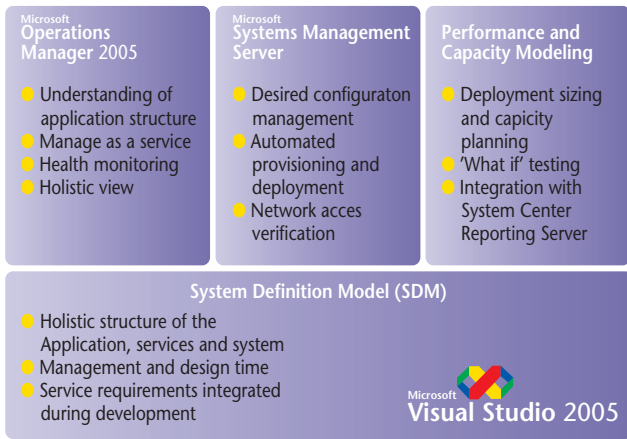
- Door alle kennis van een applicatie op te slaan in modellen, die zowel voor de ontwikkelomgevingen als voor de systeem management infrastructuur beschikbaar zijn. Hierbij wordt niet alleen de informatie over applicatie vastgelegd in modellen, maar ook de operationele requirements en infrastructuurle aspecten.
- Door gebruik te maken van de eigen System Management producten (onder meer SMS en MOM) en deze producten uit te breiden ten bate van datacenter automation.
- Door gebruik te maken van virtualisatie voor besturingssystemen en storage. Met name voor Datacenter Automation is virtualisatie een van de meest belangrijke componenten.

Microsoft hoopt met het Dynamic System Initiative de concurrentie van zich af te houden (open source) en marktaandeel te verwerven in de top van de markt, maar dat kan alleen als de klant profijt ziet in het gebruik van de Microsoft middelen. En dat zou volgens Microsoft moeten zijn:

- betere benutting van infrastructuur
- verlagen van beheerkosten en vrijmaken van budget voor andere doelen
- snellere ontwikkeltijd en minder tijd gemoeid met onderhoud
- het kunnen voldoen aan bedrijfs- en IT-richtlijnen (compliance)
- betere afstemming met de business en sneller kunnen reageren op de (gewijzigde) wensen van de business.

DSI NADER BEKEKEN In de basis is DSI een raamwerk dat er voor zorgt dat de complete lifecycle van een IT-systeem in een gecontroleerde vorm gecreëerd, gedistribueerd en beheerd kan worden. Dit betekent dat bij de ontwikkeling van systemen reeds rekening gehouden moet worden met de omgeving waarin het systeem gaat werken, de standaarden en richtlijnen die gelden in de betreffende omgeving en de criteria die gelden om de gebruikerservaring te optimaliseren. Concreet vertaalt dit zich naar een drietal aandachtsgebieden:

- *Design*
De eerste stap is het bouwen van software met tools die het ontwikkelaars mogelijk maken om tijdens de ontwikkelingen samen te werken binnen een gedefinieerde omgeving die rekening houdt met topologie, de configuraties hiervan en de benodigde hardware



AFBEELDING 2. Het System Definition Model als onderliggende technologie

resources. De eerste stap hiertoe is door Microsoft genomen met de in maart 2006 verschenen Visual Studio 2005 Team Edition for Architects.

- *Deployment*

Om op een gecontroleerde manier de ontwikkelde software te distribueren naar de gewenste omgeving zijn mechanismen nodig die rekening houden met de tijdens het ontwikkelproces gebruikte methodieken. Deze kennis en technologie wordt door Microsoft beschikbaar gesteld in Windows 'Longhorn Server' en de nieuwe versie van Systems Management Server (v4).

- *Operations*

Nadat een systeem in gebruik genomen is dienen de kennis en technieken die vastgelegd zijn bij het ontwikkelen en distribueren vertaald te worden naar operationele managementprocessen. Hiervoor zijn tools nodig die real-time feedback geven over de gebruikerservaring, geautomatiseerd beheertaken uitvoeren en hardware-resources aanspreken op basis van gesignaleerde behoeften. De volgende versie van Microsoft Operations Manager (MOM v3) en Windows 'Longhorn' Server bieden hiervoor een concrete oplossingrichting.

SYSTEM DEFINITION MODELLING Het belangrijkste aspect van IT Lifecycle Management is de integratie van de verschillende fases van de levenscyclus van een applicatie. Voor het integreren van de verschillende fases heeft Microsoft een soort supermodel neergezet, namelijk SDM. Microsoft omschrijft SDM als volgt:

'SDM is a model that is used to create definitions of distributed systems. Simply defined, a distributed system is a set of related software and hardware resources working together to accomplish a common function.'

SDM is dus een blauwdruk van het gehele systeem. Alle producten (VSTS, MOM, SMS) maken gebruik van dit model en voegen er informatie aan toe.

In het SDM-model wordt door het ontwikkelgeregenschap de informatie over een systeem opgeslagen, zoals bijvoorbeeld uit welke software- en hardware componenten het systeem bestaat. Het model bevat alle informatie die nodig is om een gedistribueerd systeem uit te rollen en te beheren, maar ook om het datacenter dynamisch te configureren.

De ambities zijn hoog want er wordt gesproken over server-, storage- en netwerkconfiguratie. Om dit te kunnen realiseren zal het datacenter gevirtualiseerd moeten worden en wel inclusief het netwerk.

Het SDM model wordt onder meer gebruikt tijdens de operatie voor health-monitoring (functioneert het systeem en de onderliggende infrastructuur zoals van tevoren aangegeven). Functioneert een systeem niet naar behoren door onvoldoende resources dan worden er automatische correctieve acties genomen. Maar ook zaken als policy-management en SLA-management zouden hiermee ingevuld worden. Doelstelling is dat de systemen 'self-managing' worden.

Naast de bovenstaande functionaliteit gaat DSI bieden:

- De mogelijkheid om uitrolscenario's virtueel te testen voor een daadwerkelijke implementatie
- De mogelijkheid om eenvoudige SDM-modellen te creëren op basis van de bestaande applicaties en infrastructuur.

SDM is een op Simple Object Access Protocol (SOAP) gebaseerde modellerings-taal die gebruikt kan worden om abstracte, dynamische modellen te creëren van gedistribueerde systemen. Een SDM-model beschrijft de complete structuur van een systeem: de logische opbouw, de relaties met andere systemen, de afhanke-

SDM is dus een blauwdruk van het gehele systeem

lijkheden, het gewenste gedrag en de constraints waarbinnen het systeem kan functioneren. Belangrijk hierbij is dat een SDM-model een levend geheel is dat gedurende de lifecycle van een systeem mee evolueert:

- *Design*

Tijdens het ontwerpen en ontwikkelen van een systeem worden basieke regels en configuraties gedefinieerd.

- *Deployment*

Wanneer het systeem gedistribueerd wordt, worden gedetailleerde configuratie gegevens, constraints en afhankelijkheden toegevoegd.

- *Operations*

Wanneer het systeem in gebruik is worden er best practices ontwikkeld en toegevoegd aan het model.

DESIGNING De eerste fase van de lifecycle van systemen waar DSI zich in manifesteert is de design fase. In het onlangs uitgebrachte Visual Studio Team System (VSTS) software lifecycle platform heeft Microsoft een aparte editie opgenomen om architecten te ondersteunen: Visual Studio 2005 Team Edition for Architects. Binnen deze editie is het voor architecten mogelijk om grafische blauwdrukken te maken van de te ontwikkelen solution en deze te matchen met een grafische weergave van de omgeving waarin de solution moet gaan functioneren.

Binnen VSTS worden twee architect rollen onderscheiden: de Infrastructure Architect en de Solution Architect. De Infrastructure Architect is verantwoordelijk voor het modelleren van de infrastructuur waarin het systeem gaat functioneren. Binnen VSTS heeft de Infrastructure Architect hiervoor de beschikking over de Logical Datacenter Designer voor het creëren van het model. In dit model wordt vastgelegd uit welke hardware, zones, endpoints, hosts en security constraints de desbetreffende omgeving bestaat. Het maken van dit model zorgt er voor dat de te ontwikkelen solution binnen de constraints van de infrastructuur gedistribueerd kan worden. Deze verificatie en controle van de ontwik-

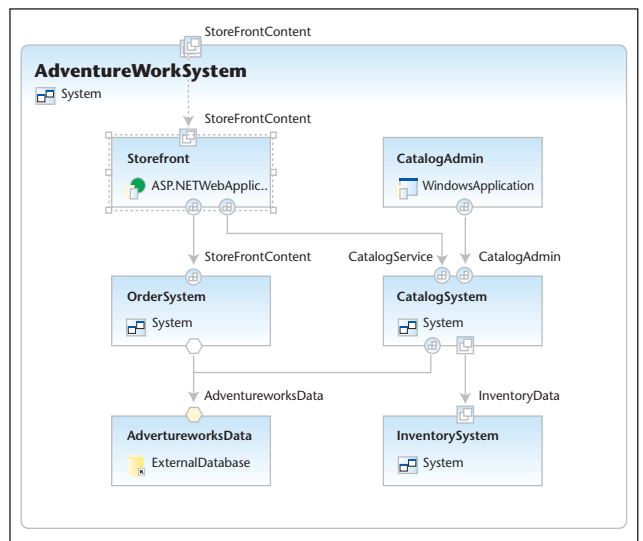
Binnen VSTS worden twee architectrollen onderscheiden

keling van een solution binnen een infrastructuur noemen we Design for Deployment.

De tweede architectrol is de Solution Architect. Deze rol is verantwoordelijk voor het modelleren van de solution en speelt logischerwijs een grotere rol binnen VSTS. In dit model wordt een abstracte weergave gebouwd van de solution-architectuur. In dit model wordt bijvoorbeeld opgenomen; het type solution, de relaties en connecties tussen componenten en constraints. De Solution Architect wordt met de tools in VSTS in staat gesteld om te werken met Domain Specific Languages (DSL's): abstracte weergaves van solution-componenten op basis waarvan code gegenereerd en gedelegeerd kan worden.

Voor het faciliteren van deze twee architectrollen zijn er een aantal design-tools in VSTS opgenomen; de Distributed System Designers:

- **Logical Datacenter Designer**
Hiermee wordt de Infrastructure Architect in staat gesteld het model van de infrastructuur te creëren: het Logical Datacenter model.
- **Application Designer**
De Application Designer stelt Solution Architects in staat het model van de solution-architectuur te definiëren. Het model dat hierbij gecreëerd wordt kan



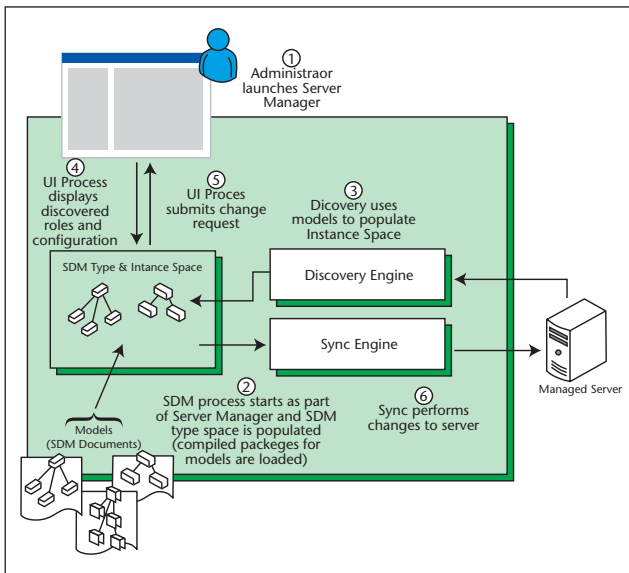
AFBEELDING 3. Een e-commerce website gemodelleerd in de Logical Datacenter Designer

gecontroleerd worden tegen het Logical Datacenter model om te verifiëren dat de solution matcht met de infrastructuur.

- **System Designer**
Met deze designer kan de configuratie van de applicatie gemodelleerd worden. Wanneer bijvoorbeeld één gemodelleerde solution binnen twee verschillende datacenters gedistribueerd moet worden, kan met de System Designer de benodigde configuraties ontworpen worden.
- **Deployment Designer**
Hiermee kunnen gebruikers een proef-deployment doen waarbij een validatie uitgevoerd kan worden van de deployment van de solution in het Logical Datacenter model. Hierbij kan een SDM-rapport gegenereerd worden van conflicten.

De met Visual Studio Team System beschikbaar gekomen technieken zorgen ervoor dat er tijdens het ontwikkelen van applicaties al rekening gehouden wordt met de verdere lifecycle van de applicatie. Dit zorgt er voor dat de transitie van ontwikkeling naar ingebruikname in een meer gecontroleerde vorm gebeurt en is een belangrijke eerste stap in de DSI-filosofie.

DEPLOYING Nadat een applicatie is opgeleverd kan binnen de nieuwe versie van SMS; System Center Configuration Manager 2007 (SCCM2007) de distributie naar de infrastructuur verzorgd worden. Dit gebeurt aan de hand van de bij de applicatie geleverde SDM modellen. Doordat tijdens de ontwikkeling van de applicatie rekening is gehouden met de infrastructuur zijn de relevante data als positionering, afhankelijkheden en beperkingen bekend. Dergelijke best-practices die de optimale omgeving beschrijven waarin de applicatie fungeert, vatten we



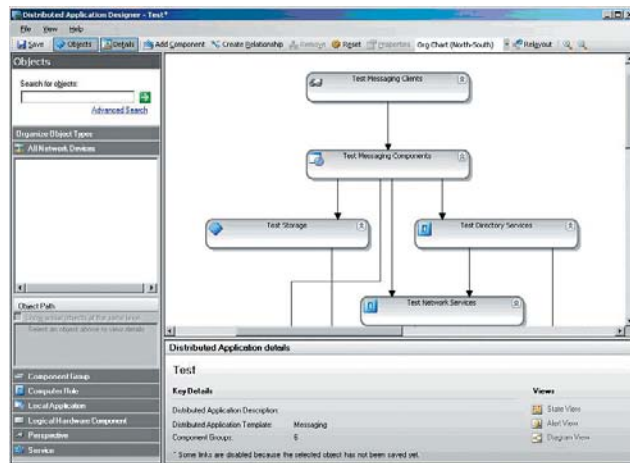
AFBEELDING 4: De Server Manager console in Windows "Longhorn" Server

onder de term: Desired Configuration Management (DCM).

Nadat SCCM2007 een applicatie gedistribueert heeft blijven de SDM modellen van de applicatie een blijvende blauwdruk om de Desired Configuration te onderhouden. Dit is een waardevolle aanvulling op de huidige capaciteiten van SMS waarbij de nadruk vooral ligt op het gedetailleerd inventariseren van hardware en software. Wijzigingen in de onderliggende infrastructuur die de applicatie raken kunnen door vergelijking met de best-practice DCM modellen bijvoorbeeld gesignaleerd worden en vervolgens gecontroleerd opgelost. Distributie van applicaties wordt op deze manier een continue proces dat de complete lifecycle van de applicatie volgt.

Behalve bij applicaties zal ook de distributie en installatie van Windows componenten op basis van SDM modellen gebaseert worden. In Windows "Longhorn" Server fungeert de Server Manager console bijvoorbeeld als GUI om met de SDM modellen van serverrollen te communiceren. Dit zorgt ervoor dat een beheerder grafisch met wizards kan werken om onderliggende meer gecompliceerde services en afhankelijkheden met behulp van SDM kan configureren en installeren. Na installatie blijft het ook hier mogelijk om via de Server Manager console de Desired Configuration van de rollen te monitoren.

OPERATIONS Nadat de applicatie of serverrol in de infrastructuur is opgenomen breekt de beheerfase aan. Dit is de fase die over het algemeen qua tijdsbesteding en kosten de grootste investering kost. Tijdens de ontwikkeling van de toepassing zijn parallel de SDM modellen ontwikkeld die de beheerprocessen technisch



AFBEELDING 5. De Distributed Application Designer in SCOM2007.

en functioneel ondersteunen. Deze worden bij de toepassing meegeleverd om in de beheertools opgenomen te worden.

Het door ontwikkelteams bijleveren van operational best-practices, monitoring triggers, etc. kennen we al van de Management Packs die voor Microsoft Operations Manager 2003 en 2005 uitgebracht zijn. Met de nieuwe (en hernoemde) versie van MOM: System Center Operations Manager 2007 (SCOM2007) wordt voor het eerst management van IT-componenten op basis van SDM modellen geboden. Dit biedt beheerders ongekende nieuwe mogelijkheden waarvan de belangrijkste end-to-end service monitoring is.

Omdat de bij een gedistribueerde applicatie geleverde SDM modellen alle relevante entiteiten omvatten is het mogelijk om applicaties op functionaliteit te monitoren. Dit houdt in dat een beheerder niet langer de onderliggende hardware, services, verbindingen, etc. afzonderlijk hoeft te behandelen maar de applicatie end-to-end als één geheel kan monitoren en beheren. Hierbij is een beheerder niet afhankelijk van bijgeleverde SDM modellen maar kunnen deze ook zelf gemaakt worden in de in SCOM2007 geïntegreerde Distributed Application Designer. Deze stelt beheerders in staat om grafisch de componenten van een applicatie in kaart te brengen en deze op te slaan als een SDM model. Wanneer dit eenmaal gedaan is kan het model op de standaard wijze bewaakt worden.

Om de functionaliteit van een applicatie op een representatieve manier te bewaken kan er binnen SCOM2007 gebruik gemaakt worden van Synthetic Transactions. Hierbij wordt een dummy transactie gecreeerd dat zich gedraagt als een gebruiker, het systeem van begin tot eind doorloopt en op basis hiervan statistieken rapporteert en events genereert.

TOEKOMSTIGE TOEPASSINGEN Naast de toepassing van de DSI principes in nieuwe versies van bestaande

producten kondigde Microsoft eerder dit jaar ook een product aan voor de servicedesk-markt; codenaam "Service Desk". Deze applicatie is er op gericht om op MOF en ITIL gebaseerde workflows in beheerorganisaties te faciliteren en zal onderdeel uit gaan maken van de System Center familie. Naast een workflow engine wordt speciale aandacht besteedt aan de CMDB (Configuration Management Database). Deze zal in staat zijn om via de SDM modellen de aanwezige infrastructuur componenten in kaart te brengen. Daarnaast belooft Microsoft een hechte integratie met o.a. Office 2007, SCCM2007 en SCOM2007. De release van "Service Desk" wordt eind 2007 verwacht.

Bij de initiële release van Windows "Longhorn" Server zal de SDM engine alleen actief zijn tijdens gebruik van de Server Manager. De volgende stap zal zijn om de SDM engine als standaard aanwezige Windows service op te nemen. Hiervoor zal er lokaal een SDM-store gedefinieerd worden die door de SDM service dynamisch gepopuleert wordt. De SDM-store bevat altijd up-to-date gegevens en wordt het primaire koppelveld voor het op afstand uitlezen en muteren van de aanwezige SDM modellen. De eerder genoemde CMDB functie in het aangekondigde servicedesk product is een voorbeeld van het toepassen van deze functionaliteit.

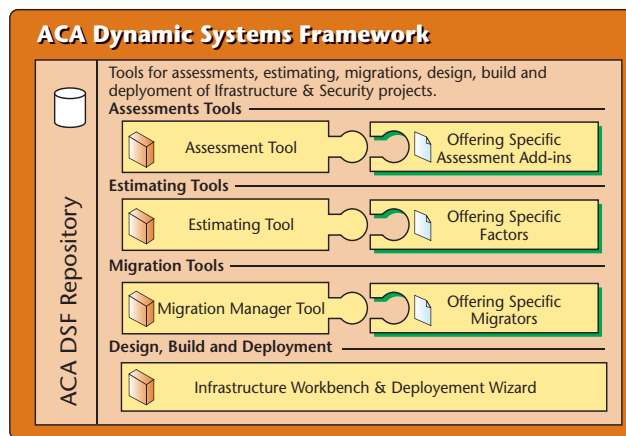
DATA CENTER AUTOMATION Datacenter Automation is niet nieuw in de IT wereld. Eerder dan Microsoft hadden IBM (Autonomic Computing/On Demand Computing), SUN (N1) en HP (Adaptive Infrastructure) initiatieven op dit vlak gelanceerd. In de literatuur wordt over het algemeen de term Autonomic Computing gebruikt in plaats van Datacenter Automation, de term die Microsoft hanteert.

Autonomic Computing houdt in dat systemen zichzelf moeten gedragen als een zelfregulerend biologisch systeem. De term autonoom is een begrip uit de biologie: "ons autonoom werkend zenuwstelsel bewaakt onder meer de hartslag, controleert de bloedsuikerspiegel en houdt de lichaamstemperatuur constant".

Autonomic Computing is een zelfhelende, zelfbeheerende, op regels (policy's) gebaseerde infrastructuur. Een autonoom systeem is:

- self-configuring – in staat om zichzelf te installeren
- self-healing – in staat om verstoringen (pro-actief) te lijf te gaan
- self-protecting – in staat om zichzelf te beschermen tegen technische storingen, virussen e.d.
- self-optimizing – in staat om dynamisch resources te alloceren om processen zo efficiënt mogelijk te laten uitvoeren

De benodigde informatie om bovenstaande te realiseren haalt de systeem management infrastructuur uit



AFBEELDING 6: ACA DSF als geconsolideerde repository

het bij de systemen meegeleverde System Definition Model.

Microsoft stopt veel energie en middelen in Autonomic Computing en denkt hiermee concurrentievoordeel te kunnen halen. Daarnaast heeft Microsoft de essentiële middelen in handen om Autonomic Computing voor het Windows platform te realiseren. Deze middelen zijn Windows Server, Microsoft Operations Manager (MOM), System Management Server (SMS), Virtual Server en System Center.

Virtualisatie is een zeer belangrijk onderdeel van de strategie om self-managing systemen te kunnen realiseren. Virtualisatie maakt het mogelijk om beschikbare resources dynamisch toe te wijzen en te configureren. Virtualisatietechnologie wordt toegepast op tal van gebieden, van besturingssystemen tot opslagsystemen. Doel is om het gehele datacenter te virtualiseren, zodat het bijvoorbeeld mogelijk wordt om nieuwe virtuele fysieke machines toe te voegen aan het datacenter.

Het is dan ook niet verwonderlijk dat Microsoft veel energie steekt in virtualisatie. De nieuwste versie van Virtual Server (VS 2005 R2) biedt al een aantal aspecten van autonomic computing, zoals 'automatic load balancing', 'dynamic updating' en correctieve acties voor verstoringen ('unplanned downtime').

Doelstelling van Microsoft is om de systeem management producten binnen afzienbare tijd virtualisatie te laten ondersteunen. MOM 2005 biedt al beheermogelijkheden voor virtuele servers en sessies.

KANSEN VOOR THIRD PARTIES De integratie van de DSI-concepten in Microsoft-producten biedt volop kansen voor third party's; ontwikkeling van op SDM gebaseerde management packs en DCM-configuraties. Aansluiting op DSI geeft third party's de mogelijkheid zich te profileren als consumentvriendelijke partij die betrokkenheid toont door bijvoorbeeld ontwikkeling van applica-

ties volgens het 'Developing for Operations' principe uit te voeren.

Avanade is als joint venture tussen Accenture en Microsoft zeer intensief bezig om additionele 'assets' te ontwikkelen binnen het DSI framework. Eén van de belangrijkste Avanade Assets hierbij is ACA DSF ofwel Dynamic Systems Framework.

ACA DSF is een framework dat Avanade-consultants in staat stelt om Microsoft oplossingen op basis van abstracte modellen te implementeren. Hiervoor worden wereldwijd binnen de Avanade-communities modellen van bestaande en toekomstige Microsoft-oplossingen ontwikkeld. In deze modellen worden de Microsoft best-practices en Avanade's unieke kennis gebundelt en opgeslagen. Nadat de rollen centraal op kwaliteit gekeurd zijn worden ze wereldwijd beschikbaar gesteld voor (her-) gebruik door Avanade consultants. Vervolgens kan een Avanade consultant via een grafische interface de modellen bewerken om zo de daadwerkelijke implementatie uit te voeren. Hierbij kan gekozen worden om voor de distributie aan te sluiten op bestaande en toekomstige technologieën (Automated Deployment Services, SCCM2007) of voor DSF's ingebouwde deployment engine.

Binnen ACA DSF is de lifecycle van Microsoft producten voor een system integrator volgens de DSI-principes vastgelegd. Dit zorgt ervoor dat bestaande en toekomstige Microsoft-producten door Avanade bij klanten als beheersbare oplossingen geïmplementeerd worden. Behalve de voordelen die dit biedt voor de overdracht naar beheer betekent dit ook dat de controle over implementatie sterk vergroot wordt.

Door de unieke kennis van Avanade in ACA DSF te modelleren en wereldwijd te delen kunnen Avanade-consultants en klanten direct profiteren van de door DSI geboden principes.

Sterke punten van ACA DSF:

- Ontwerpen van infrastructuren en datacenters
- Validatie tegen Microsoft en Avanade Best Practices
- Design here, roll-out everywhere
- Grotere controle over en tijdens de implementatie van infrastructuur componenten
- Automatisch genereren van documentatie
- Gebaseert op de principes van het Dynamic Systems Initiative

CONCLUSIE Het moge duidelijk zijn dat Microsoft met DSI en daarin SDM de goede weg is ingeslagen met betrekking tot het managen van verschillende IT-lifecycles zoals Application Lifecycle Management en Infrastructure Lifecycle Management. Behalve Microsoft bieden ook derde partijen zoals Avanade interessante oplossingen aan met betrekking tot het invullen van het DSI framework.

Met de beschikbaarheid van Visual Studio Team System begin 2006 is een eerste belangrijke stap gezet in de praktische toepasbaarheid van DSI. Met de aankomende golf van Microsoft-producten komt de volledige functionaliteit van DSI beschikbaar. Om hier op voorbereid te zijn is het nu de tijd om met DSI te starten, dit kan redelijk eenvoudig:

- Implementeer Microsoft MOM 2005 en monitor elk Windows serverplatform en applicatie.
- Implementeer SMS en WSUS om uw infrastructuur up-to-date te brengen en te houden
- Train uw IT staff in MOF of ITIL
- Zorg dat uw in-house development omgeving overstapt naar Visual Studio Team System
- Adopteer de best practices voor development en managability van uw omgeving en applicaties.
- Contacteer Avanade hoe u dit het beste kunt uitvoeren met behulp van hun assets.

Referenties

- Microsoft Dynamic Systems Initiative:
www.microsoft.com/windowsserversystem/dsi/
- System Definition Model Overview:
www.microsoft.com/windowsserversystem/dsi/sdmwp.msp
- Microsoft Visual Studio Team System:
<http://msdn.microsoft.com/vstudio/teamsystem/>
- Microsoft System Center:
www.microsoft.com/systemcenter/

*André Akkerman – Infrastructure Consultant Avanade
(andrea@avanade.com) Aram Smith – Infrastructure Architect
Avanade (arams@avanade.com). Met dank aan: Morten Finnerud –
Enterprise Architect Reaal verzekeringen.*