

MASTER

Kwaliteitsraamwerk voor Enterprise Application Integration

Hoogendoorn, Mark

Award date:
2002

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

TECHNISCHE UNIVERSITEIT EINDHOVEN

Faculteit Wiskunde en Informatica

AFSTUDEERVERSLAG

Kwaliteitsraamwerk voor
Enterprise Application Integration

door
M. Hoogendoorn

Afstudeerdocent:
dr. ir. G.J.P.M. Houben

15 April 2002

pwc

Afstudeer verslag
Mark Hoogendoorn

E.A.I – Enterprise Application Integration

1 Voorwoord

Dit verslag dient als afsluiting van mijn studie technische informatica aan de Technische Universiteit Eindhoven. De afgelopen maanden heb ik bij PricewaterhouseCoopers gekeken naar de relatie tussen kwaliteit en software en getracht een kader te ontwikkelen om meer vat op deze materie te krijgen. Ik wil graag mijn afstudeer begeleider van de TU/e de heer dr. ir. G.J.P.M. Houben bedanken voor zijn ondersteuning en uitleg. Daarnaast gaat voor zijn kritische blik en aanbevelingen op het verslag een woord van waardering uit naar de tweede TU/e begeleider de heer dr. M.R.V. Chaudron. Daarnaast wil ik graag de bedrijfsbegeleider, de heer RE W. Ehren zeer hartelijk bedanken voor zowel zijn op- en aanmerkingen alsook de ondersteuning in de moeilijke perioden. Uiteraard wil ik ook mijn familie en in het bijzonder mijn vriendin, thans verloofde, veel dank betuigen voor hun geduld en ondersteuning.

2 Inhoudsopgave

AFSTUDEER VERSLAG	1
1 VOORWOORD	2
2 INHOUDSOPGAVE	3
3 INLEIDING	5
3.1 INLEIDING	5
3.2 PRICEWATERHOUSECOOPERS	5
3.3 OPERATIONAL AND SYSTEMS RISK MANAGEMENT	5
3.4 AUDITING	6
3.5 PROBLEEMSTELLING	6
3.6 LEESWIJZER	8
4 INTEGRATIE	9
4.1 INLEIDING	9
4.1 INTEGRATIE MOGELIJKHEDEN	9
4.2 INTEGRATIE KEUZES	10
4.2.1 'POINT TO POINT'	10
4.2.2 ERP-GEBASEERDE INTEGRATIE	11
4.2.3 MIDDLEWARE GEORIËNTEERDE AANPAK	12
4.3 INTEGRATIE MECHANISMEN	13
4.3.1 CALL INTERFACE	13
4.3.2 MESSAGING	14
4.3.3 DATA ACCESS/FILE TRANSFER	14
4.4 CONVERSIE EN TRANSFORMATIE	15
5 DE ONTSTAANSREDEN VAN EAI	16
5.1 INLEIDING	16
5.2 DE ONTWIKKELING VAN ERP PAKKETTEN	16
5.3 DE ONTWIKKELING VAN E-BUSINESS	17
5.4 OVERIGE APPLICATIES	18
5.5 DRIJFVEREN VOOR E.A.I	20
6 KWALITEIT	22
6.1 INLEIDING	22
6.2 DOMEINEN VAN IT AUDITING	22
6.3 KWALITEITSASPECTEN	23

6.4	KWALITEIT EN EAI	24
7	ARCHITECTUUR	26
7.1	INLEIDING	26
7.2	HET ONTWERPEN VAN EEN ARCHITECTUUR VAN EEN GEÏNTEGREERD SYSTEEM	27
7.2.1	DE VERSCHILLENDE TYPEN ARCHITECTUUR	27
7.2.2	RAAMWERKEN	28
7.3	ARCHITECTUURKENMERKEN	28
7.3.1	ARCHITECTONISCHE FUNDERING	29
7.3.2	BASIS KENMERKEN	30
7.3.3	GEAVANCEERDE KENMERKEN	31
7.3.4	MANAGEMENT	32
7.3.5	ROBUUSTHEID	33
7.3.6	PERFORMANCE	34
7.3.7	COMPONENT RELATIES	34
7.3.8	EXTERNE FACTOREN	35
7.4	CONTROLE RAAMWERK	36
8	DE BOUWSTENEN VAN EAI	37
8.1	INLEIDING	37
8.2	DE VERBINDINGSARCHITECTUUR	37
8.3	ADAPTOREN EN CONNECTOREN	39
8.4	WORKFLOW MANAGEMENT	41
9	EEN RAAMWERK VOOR EAI	42
9.1	INLEIDING	42
9.2	REFERENTIE MODEL VOOR EEN EAI COMPONENT	42
9.3.1	TRANSPORTLAAG	42
9.3.2	DE TRANSACTIE LAAG	43
9.3.3	DE TRANSFORMATIE LAAG	43
9.3.4	TIMING	43
9.3.5	DE PROCES-LAAG	44
9.4	DIENSTENMODEL	44
9.4.1	HET ONTWERP VAN HET MODEL	45
9.4.2	DE COMPONENTEN	47
9.4.3	KWALITEITSASPECTEN IN RELATIE TOT DE COMPONENTEN	50
9.4.4	HET MODEL IN DE PRAKTIJK.	52
9.4.5	CONCLUSIE EN VERVOLGONDERZOEK	54

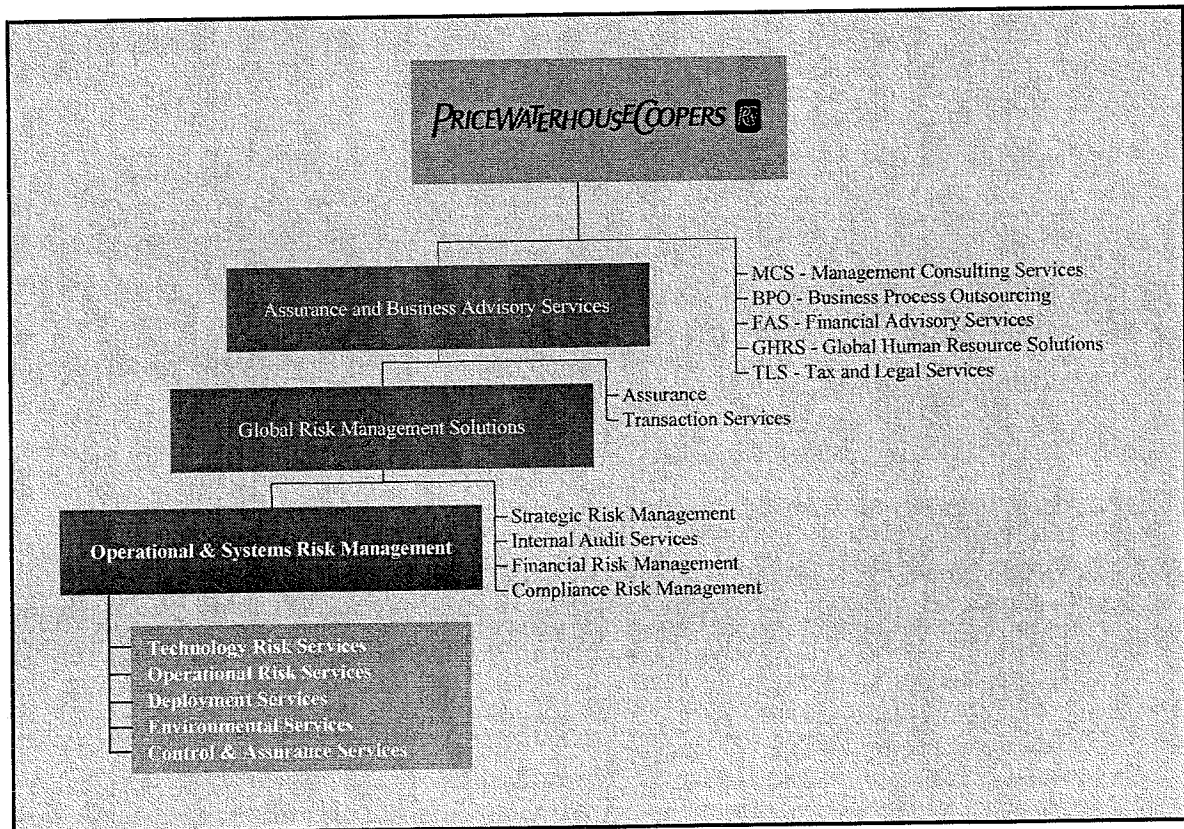
3 Inleiding

3.1 Inleiding

In dit hoofdstuk wordt een korte beschrijving gegeven van de stagebiedende organisatie PricewaterhouseCoopers en de business unit waarin ik werkzaam ben. Daarna zal worden ingegaan op de probleemstelling van dit onderzoek.

3.2 PricewaterhouseCoopers

PricewaterhouseCoopers is in Juni 1998 ontstaan uit de fusie tussen Pricewaterhouse en Coopers & Lybrand. Deze fusie heeft geleid tot de grootste accountants- en adviseursorganisatie ter wereld. Binnen PricewaterhouseCoopers zijn 150000 professionals werkzaam in 152 verschillende landen die een totale omzet genereren van 32 miljard gulden. PricewaterhouseCoopers heeft in Nederland 26 vestigingen en 5500 medewerkers. De accountants, belastingadviseurs en management consultants werken multidisciplinair samen in opdracht van nationale en internationale, kleine, middel en grote ondernemingen en organisaties. Het organisatie schema van PricewaterhouseCoopers ziet er als volgt uit:



Afbeelding 1

Bij PricewaterhouseCoopers is het gebruikelijk om kennis over sectoren zoveel mogelijk te delen en tegelijkertijd te clusteren. Hiervoor worden zogenaamde sectorgroepen gevormd, welke specifieke kennis hebben van een industry. Een sectorgroep komt op twee niveaus voor, namelijk landelijk per business unit (bijvoorbeeld Operational and Systems Risk management).

3.3 Operational and Systems Risk Management

Operational and systems risk management (OSRM) houdt zich bezig met het beoordelen van de kwaliteit van de informatievoorziening in een geautomatiseerde omgeving, en het beheersen van de daarmee samenhangende risico's. De unit is gevestigd in Amsterdam, Den Haag, Eindhoven, Groningen, Rotterdam en Utrecht. In totaal werken er ongeveer 350 personen.

De beoordeling van de kwaliteit van de informatievoorziening vindt plaats op een drietal gebieden, welke worden aangeduid met Service Lines. Deze Service Lines zijn:

- Control & Assurance services (CAS) richt zich met name op de kwaliteit van de interne controle maatregelen zoals die binnen een organisatie worden gehanteerd. Dit zijn klassieke EDP-audit werkzaamheden.
- Operational Risk Services (ORS) helpt de klant om de ICT-gerichte risico's te beheersen, welke worden gelopen bij het uitvoeren van de core business activiteiten.
- Technology Risk Services (TRS) ondersteunt bedrijven bij het bepalen van risico's bij het ontwikkelen, beheersen en besturen van innovatieve Informatie- en communicatie technologie binnen de organisatie. Hierbij staat het innovatieve karakter centraal. Een voorbeeld van een innovatieve IT-ontwikkeling voor veel bedrijven is Electronic Commerce. TRS kan de klant adviseren op dit gebied, voor wat betreft de risico's van E-Commerce

Om klanten van PricewaterhouseCoopers optimaal van dienst te kunnen zijn is er een indeling gemaakt naar zogenaamde industries. De volgende Industries worden onderscheiden.

- Services
- Telecommunication, Information, Communication and Entertainment
- Financial Services
- Middle Market
- Consumer and Industrial Products
- Energy and Mining

3.4 Auditing

Een van de kernactiviteiten van OSRM is het auditten van informatie systemen. Informatie systeem auditing is het verzamelen en beoordelen van bewijzen om te bepalen of een computer systeem de activa beschermt, de data integriteit waarborgt, het toelaat dat de doelen van organisatie effectief worden behaald en dat de bedrijfsmiddelen efficiënt worden gebruikt.

De behoefte naar IT-audits is ontstaan door een aantal factoren. Deze zijn in te delen in de volgende groepen:

- Organisatie kosten voor data verlies
- Kosten van het nemen van verkeerde beslissingen
- Kosten van computer misbruik
- Hoge kosten van computer fouten
- Het waarborgen van de privacy
- Gecontroleerde evolutie van computer gebruik
- Waarde van hardware, software en personeel

De doelen die door een organisatie worden nagestreefd zijn onder te verdelen in de volgende vier groepen:

- Betere beveiliging van de activa
- Verbeterde data integriteit
- Verbeterde effectiviteit van het systeem
- Verbeterde efficiency van het systeem

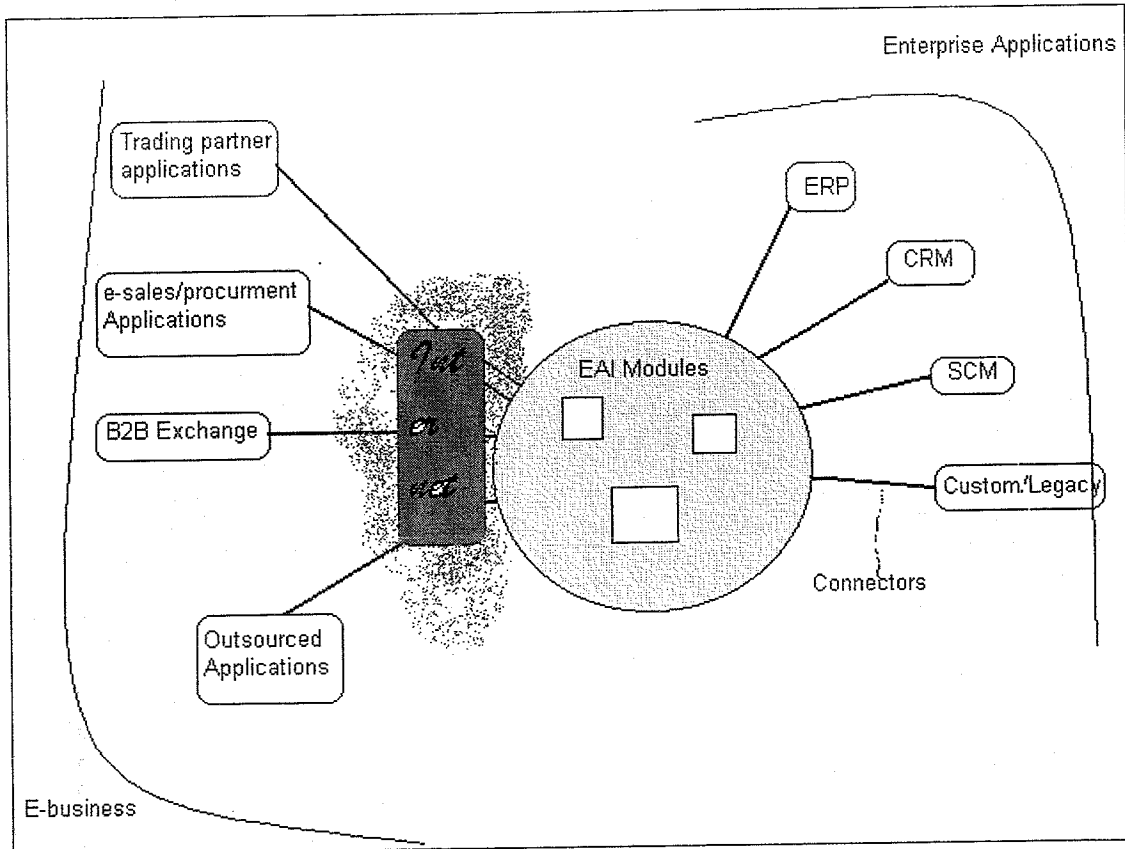
Een audit resulteert in een waardeoordeel over de werking van een systeem (bijvoorbeeld over de correcte verwerking van data). Dit waardeoordeel komt tot stand aan de hand van een vooraf vastgestelde verzameling kwaliteitseisen.

3.5 Probleemstelling

Enterprise application Integration (EAI) beschrijft een nieuwe categorie software producten die de integratie van bestaande en nieuwe applicaties mogelijk maakt en coördineert.

EAI gaat verder dan bijvoorbeeld het koppelen van data bestanden of reeds bestaande middleware oplossingen zo worden ook objecten en processen aan elkaar gekoppeld, zodanig dat een stuk bedrijfslogica in het EAI systeem verwerkt zit.

Enterprise application integration vervangt de complexiteit van 'point to point' integratie door een centraal integratie geraamte te creëren (zie afbeelding 2).



Afbeelding 2

Dit centrale geraamte neemt meerdere taken op zich. Men moet hierbij denken aan message oriented middleware, dataverwerking en -routing en bedrijfsproces automatisering.

Dit heeft geleid tot volgende probleemstelling :

De aard, het aantal en de omvang van systemen in ondernemingen nemen razendsnel toe. Ontwikkelingen zijn bijvoorbeeld de invoering van ERP systemen en de laatste jaren door de opkomst van het internet E-business applicaties. Invoering van dit soort systemen is vaak gepaard gegaan met veel energie en investeringen. Weggooien doet men dus niet zo maar. Bedrijfsprocessen overschrijden steeds meer grenzen binnen en buiten de organisatie. Als gevolg hiervan ontstaat de noodzaak tot onderlinge informatie uitwisseling. Omdat geautomatiseerde systemen een steeds prominentere rol vervullen in bedrijfsprocessen is het noodzakelijk dat deze systemen in staat zijn onderling informatie uit te wisselen, ongeacht de aard en soort. Het middel om in deze informatieuitwisseling te voorzien is de koppeling. Omdat het aantal betrokken systemen en dien ten gevolge de complexiteit toeneemt wordt steeds meer gebruik gemaakt van een specifiek integratie hulpmiddel, de EAI Ook EAI is geen op zich zelfstaand systeem, maar is een verzameling componenten die op hun beurt ook weer afhankelijk zijn van onderliggende techniek. Al met al is EAI een complex geheel en een cruciale factor in de integratie van systemen. Omdat de keten zo sterk is als de zwakste schakel dienen (gelet op de afhankelijk van bedrijven van geautomatiseerde systemen) aan EAI hoge eisen te worden gesteld.

Een maatstaf voor deze eisen is het begrip kwaliteit. Om kwaliteit te kunnen meten dient het te worden opgedeeld in eenduidig gedefinieerde en meetbare aspecten.

Op basis van het voorgaande kan de volgende probleemstelling worden onderkend:

Welke kwaliteitsaspecten kunnen worden onderscheiden en in welke componenten dient een EAI te worden opgedeeld om een uitspraak over de kwaliteit van EAI te kunnen doen?

De deelvragen die volgen uit de probleemstelling zijn de volgende:

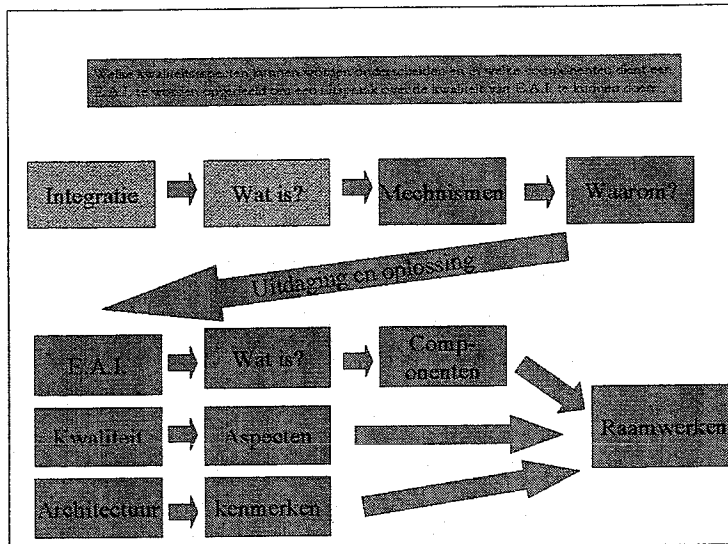
- Wat is integratie ?
- Welke mechanismen zijn er om applicaties te integreren?
- Wat is kwaliteit?
- Wat zijn de verschillende kwaliteitsaspecten?
- Welke eisen kan men stellen aan de onderliggende architectuur?
- Hoe kan men deze eisen met de kwaliteitsaspecten mappen?
- Waarom EAI?
- Wat is EAI?
- Uit welke componenten bestaat een EAI?
- Hoe kan men de kwaliteitsaspecten met de componenten mappen?

3.6 Leeswijzer

Zoals uit de deelvragen blijkt is de probleemstelling niet kort en krachtig te beantwoorden. Het te onderzoeken object moet worden gedefinieerd alsmede het subject: de kwaliteit dient handelbaar te worden gemaakt. De deelvragen zullen worden beantwoord in de volgorde zoals ze hierboven zijn opgesomd. Om de draad tijdens het lezen van dit verslag niet te verliezen, wordt gebruik gemaakt van een hulpmiddel; de zogenaamde leeswijzer. De leeswijzer is in de vorm van een stroomschema weergegeven. Telkens als een volgende deelvraag wordt beantwoord, of een object beschreven zal in dit stroomschema een andere rechthoek grijs worden gearceerd. Zo weet men altijd welk deel van het onderzoek wordt behandeld. In de volgende paragraaf is de leeswijzer opgenomen.

4 Integratie

4.1 Inleiding



Leeswijzer

Voordat wordt ingegaan op mogelijke oplossingen en de kwaliteit hiervan is het belangrijk dat het probleem van de integratie wat nauwkeuriger in kaart wordt gebracht. In dit hoofdstuk zal verder worden ingegaan op welke niveaus van integratie meestal worden onderscheiden en welke mogelijke architecturen hiervoor worden gebruikt. Daarna zullen de integratie mechanismen die momenteel worden gebruikt beschreven worden en de voor- en nadelen van bepaalde technieken worden aangeven.

4.1 Integratie mogelijkheden

Bij elk integratieproject is het van belang om eerst de gevolgen van een dergelijke verandering in kaart te brengen. Dit wordt gedaan door de integratiebreedte en -diepte aan te geven. Met integratiebreedte wordt het aantal verschillende applicaties bedoeld dat dient te worden geïntegreerd. Het integreren van een systeem dat alleen applicaties voor de inkoopfunctie koppelt is een voorbeeld van geringe *integratiebreedte*, terwijl het integreren van alle applicaties binnen de waardeketen de maximale integratiebreedte heeft.

Met de *integratiediepte* wordt een integratie graad bedoeld die zich in drie niveau's laat verdelen:

- Integratie op dataniveau
- Integratie op objectniveau
- Integratie op procesniveau

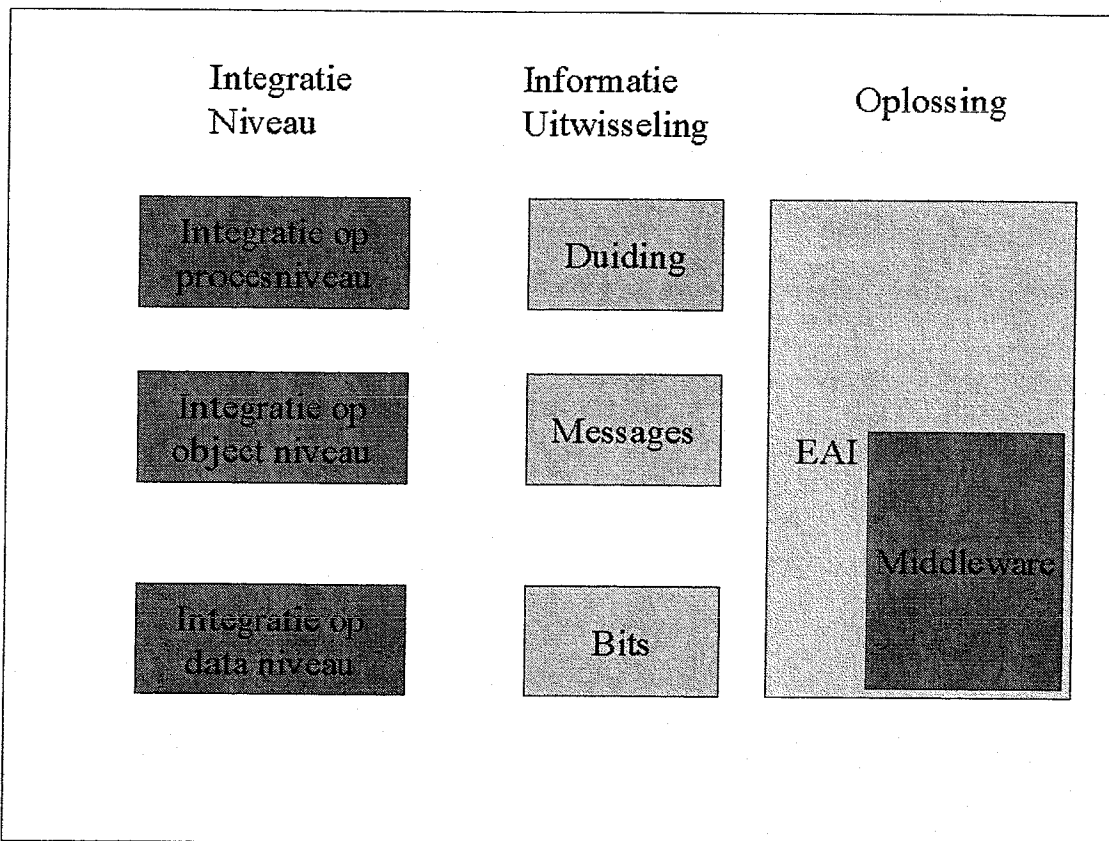
Met integratie op dataniveau wordt bedoeld dat protocollen er voor zorgen dat data op netwerk niveau van het ene systeem in het andere wordt overgedragen. Deze protocollen worden *Data Transfer Protocols* genoemd. De kern is dus verplaatsen van bytes van de ene machine naar de ander. Deze laag correspondeert met de eerste vier lagen van het OSI-model: de fysieke laag, data laag, netwerk laag en de transport laag.

Het dataniveau kan in deze samenhang als eenvoudigste vorm van integratie worden gezien. Systeem integratie kan worden vergeleken met een Brit en een Fransman die met elkaar willen communiceren. Op dataniveau wordt er voor gezorgd dat de beide gesprekspartners elkaar kunnen horen. Dit betekent echter niet dat ze elkaar ook begrijpen.

Als we deze voorstelling volgen is de volgende laag van semantische integratie er voor verantwoordelijk dat de gesprekspartners de woorden die ze uitwisselen begrijpen. In de wereld van systeemintegratie worden hiervoor objecten gedefinieerd, een object bestaat uit data en methoden. De methoden worden gedefinieerd in de interface, en deze methoden voeren bepaalde bewerkingen op de data uit. Zo zou in een

banksysteem het object *saldoInformatie* de data *RekeningNummer*, *Eigenaar*, *Saldo* etc. kunnen bevatten en door middel van de methoden *geefSaldo*, *crediteerSaldo*, etc. de data kunnen worden bewerkt.

De integratie op procesniveau is de meest uitdagende vorm van semantische integratie. Deze vorm beschrijft de ondersteuning van processen, waarbij tijdens een proces verschillende objecten worden be- en verwerkt. Voor de Brit en de Fransman betekent dit dat woorden tot zinnen worden gesmeed en deze in een bepaalde context worden geplaatst om een bepaald communicatiedoel na te streven. Er wordt in een voor beide begrijpelijke taal gecommuniceerd.



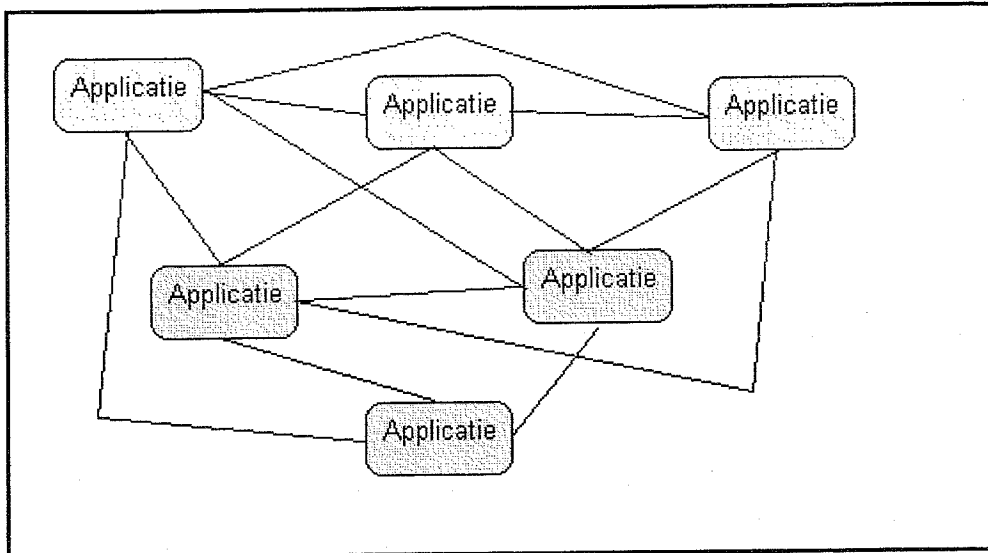
Afbeelding 3

4.2 Integratie keuzes

De integratie van interne en externe applicaties tot op het niveau van procesintegratie kan op verschillende manieren worden gerealiseerd. In de praktijk worden veel mengvormen van volgende scenario's aangetroffen, aangezien de integratie meestal niet in één keer, maar in meerdere stappen wordt uitgevoerd. Hieronder worden de meest gebruikte aanpakken weergegeven. Ten eerste het aan elkaar knopen van alle applicaties. Daarna wordt een uitbereiding van het ERP systeem besproken wordt. Afsluitend zal worden ingegaan op de middleware georiënteerde aanpak.

4.2.1 'Point to Point'

Een *point to point* verbinding is een directe verbinding tussen twee applicaties. Voor de integratie van meerdere applicaties wordt telkens een nieuwe verbinding gelegd. Hierdoor ontstaat een netwerktopologie die ook wel spaghettistructuur wordt genoemd (Zie afbeelding 4).



Afbeelding 4

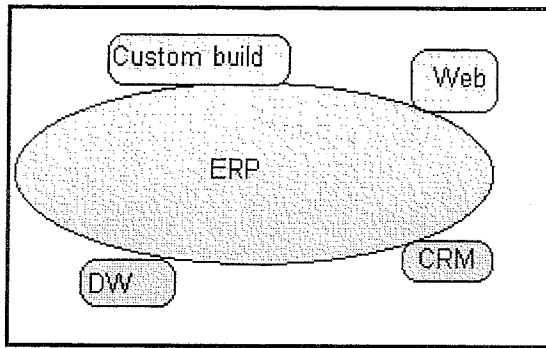
Deze architectuur is meestal het gevolg van een in de tijd gegroeide structuur (uitgebreid met nieuwe systemen en/of applicaties). Oftewel; het gegeven systeemgebied is op geen enkel tijdstip gereorganiseerd. Er werden steeds nieuwe systemen in de reeds bestaande architectuur ingebed. De belangrijkste nadelen van deze architectuur zijn:

- Veel investering in het maken van de interfaces (per verbinding 2), dit betekent dat er voor N applicaties $((N-1)N)/2$ interfaces dienen te worden ontwikkeld;
- Elke nieuw te integreren applicatie brengt veel werk met zich mee, in het geval dat de applicatie met de N bestaande moet verbonden worden: het ontwikkelen van N interfaces;
- De resources worden verre van optimaal gebruikt; In een netwerk omgeving is een knelpunt de belasting van het netwerk. Door de 'point to point' aanpak moeten de gegevens van de ene applicatie naar de andere gestuurd worden en als het meer dan twee applicaties betreft weer naar de volgende. De processoren van de individuele machines wachten te vaak ook data.
- De data is niet homogeen noch consistent; Niet homogeen omdat de data voor sommige niet bereikbaar nog interpreteerbaar is. Niet consistent omdat deze data door verschillende applicatie anders geïnterpreteerd kan worden
- Er ontstaan vertragingen bij de dataverwerking;

4.2.2 ERP-gebaseerde integratie

Er zijn veel grote stappen voorwaarts gerealiseerd bij de applicatie integratie door de invoering, van de later in dit rapport beschreven ERP (Enterprise Resource Planner) systemen (zoals Oracle, Baan, Peoplesoft, SAP en dergelijke). Deze systemen ondersteunen belangrijke bedrijfsprocessen, zoals materiaal planning en de administratie.

Er kan desondanks van worden uitgegaan dat het niet mogelijk is om meer dan 40 % [REF 1] van alle bedrijfsprocessen door ERP-systemen te ondersteunen. Branchespecifieke of ondernemingspecifieke eisen worden door speciale oplossingen uitgebreid. Daarnaast is het zo dat nieuwe aanbieders van software voor de ondersteuning van speciale processen, de mogelijkheid hebben om sneller op integratie-eisen te kunnen reageren. Het is dan ook niet te verwachten dat de ICT-eisen volledig door een ERP-systeem ooit kunnen worden gedekt. Veel eerder zullen de ERP-oplossingen tot de kern van het systeem uitgroeien. De overige individuele applicaties zullen behouden blijven en via het ERP-systeem met elkaar communiceren (Zie afbeelding 5).



Afbeelding 5

De belangrijkste nadelen van deze oplossing zijn:

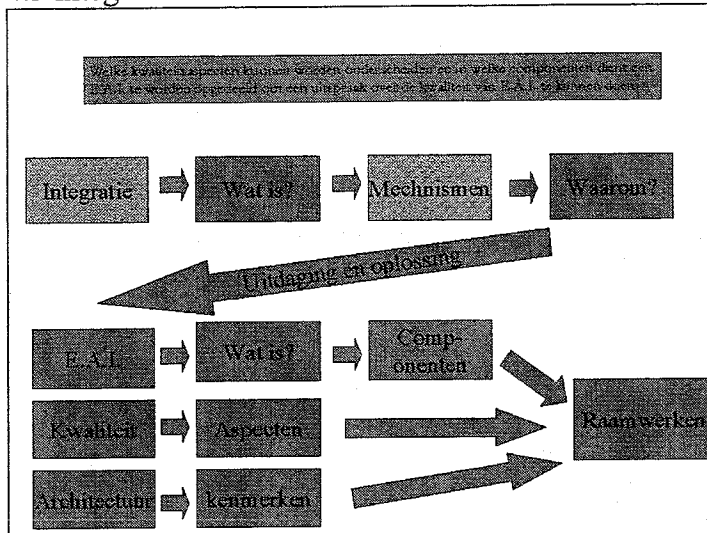
- Op ERP-gebaseerde oplossing is alleen dan voldoende, als er geen directe communicatie nodig is tussen de satellietapplicaties.
- Men zal in sommige gevallen niet de beste software kopen maar de software die met het ERP pakket samenwerkt.
- Vooral legacy kan moeilijk geïntegreerd worden , daarvoor moeten interfaces gemaakt worden.

4.2.3 Middleware georiënteerde aanpak

Middleware wordt gedefinieerd als de laag van generieke componenten tussen netwerk en applicaties. Klassieke middleware producten volgen de aanpak die de koppeling van verschillende systemen in relatief gestandaardiseerde vorm mogelijk moet maken. Middleware producten kunnen tussen twee of meerdere systemen worden geschakeld en maken het mogelijk om systemen en applicaties van verschillende makers en eventueel platform onafhankelijk, te laten communiceren en data uit te wisselen. Middleware wordt vooral in de al eerder beschreven 'point to point' middleware ingezet. De toepassing dekt vooral de integratie op dataniveau en gedeeltelijk op object niveau.

Een uitbereiding van de klassieke middleware producten zijn de Message brokers, waardoor de integratie op object niveau van meerdere systemen kan worden gerealiseerd. Hier kunnen centraal afspraken over formaten en bedrijfsregels worden vastgelegd.

4.3 Integratie mechanismen



Leeswijzer

Om bovenstaande scenario's te laten werken wordt gebruik gemaakt van een verzameling communicatie technieken. Deze zijn in te delen in drie groepen:

- Call interfaces
- Messaging
- Data access/File Transfer

Voor elk van deze groepen volgen nu enkele voorbeelden en worden de voor- en nadelen beschreven. Deze zijn belangrijk omdat ook bij het implementeren van een EAI-oplossing een keuze uit deze technieken gemaakt moet worden.

4.3.1 Call Interface

Veel applicaties bieden aanroepbare interfaces, meestal worden deze *Application Program Interface* (API) genoemd. Een API is een verzameling methoden van een applicatie die door andere applicaties gebruikt mogen worden.

Deze hoeven geen gebruik te maken van object georiënteerde oplossing zoals COM+ of CORBA, hoewel dit steeds vaker gebeurd gezien de steeds hogere eisen aan de interoperabiliteit. Voorbeelden van call interfaces zijn:

- Object interfaces
 - COM+

COM+ is een uitbreiding van het Component Object Model (COM) Microsofts modelleer gereedschap voor het ontwikkelen van applicaties. COM+ is zowel een object georiënteerde programmeer architectuur als ook een verzameling functionaliteiten van het operating systeem. COM wordt uitgebreid met een aantal componenten die het mogelijk maken meer 'event driven' te werk te gaan.

COM+ heeft als achterliggende gedachte dat het relatief makkelijk moet worden om bedrijfapplicaties die in een Windows NT omgeving draaien met de microsoft transaction server te coördineren en te ontwikkelen.
 - CORBA

Common Object Request Broker Architecture (CORBA) is een architectuur en specificatie voor het creëren, distribueren, en beheren van gedistribueerde objecten in een netwerk. Het is mogelijk programma's op verschillende locaties, gemaakt door verschillende ontwerpers, met elkaar te laten communiceren over een netwerk door een interface broker. CORBA werd ontwikkeld door de Object Management Group (OMG) en wordt herkend door de International Organization for sStandardization (ISO) als de standaard voor gedistribueerde objecten.
 - JavaBeans

Enterprise JavaBeans (EJB) is een architectuur om java programma's en componenten met elkaar te laten communiceren via het client server model.

Een component wordt een bean genoemd. Een bean is een programma dat communiceert met op de server staande software de 'servlets'.

Het voordeel van enterprise java beans ligt hem vooral in het 'client-server' idee. Bij veranderingen van de software hoeven de clients niet aangepast te worden maar alleen de op de server staande methoden die door de beans gebruikt worden.

- Transaction processing interfaces ook: TP monitoren
 - IBM Customer Information control system (CICS)
 - BEA Tuxedo
- Packaged application interfaces
 - SAPs Business API (BAPI)

SAP is een van de grootste aanbieders van ERP-pakketten. Een BAPI is een API van SAP die het mogelijk maakt SAP te koppelen aan andere applicaties. Dit betekent dat men integratie kan realiseren volgens de ERP-gebaseerde aanpak

Voordelen:

- real-time integriteit (transacties en data)
- Ingekapselde implementatie achter een algemene interface
- Men moet vanuit de bedrijfslogica redeneren en niet enkel op data niveau.
- Nieuwe componenten zijn makkelijk te integreren
- Men kan een omhulsel (wrapper) om de oudere applicaties maken en hoeft de source daarvan dus niet te veranderen om ze in het systeem op te nemen

Nadelen:

- Kan zeer complexe programmeer problemen opleveren, nieuwe technologieën gebruiken de object georiënteerde aanpak.
- Er zijn veel verschillende technologieën het reduceren van dit aantal is een van de uitdagingen voor de applicatie integratie software.
- Om synchroon gedrag te kunnen realiseren moeten de verbindingen en de applicaties tussen hen altijd actief zijn.

4.3.2 Messaging

De overdracht geschiedt door middel van berichten, de zogenaamde *messages*. De integratie door middel van *messages* wordt toegepast voor een asynchrone gedistribueerde omgeving. Met asynchroon wordt bedoeld dat er niet automatisch na het versturen van een bericht een reactie komt. Een voorbeeld van asynchrone communicatie het gebruik van e-mail.

Een bekend voorbeeld voor deze vorm van communicatie is het product MQSeries van IBM. Hierbij post een bronstelsel een data-entry in een output wachttijd, de zogenaamde queue. De applicatie MQSeries leest deze queue uit en plaatst de data-entry in de input queue van het doelsysteem. Het doelsysteem leest deze inputqueue dan uit.

Een ander voorbeeld van systemen die gebruik maken van berichten zijn 'groupware' producten zoals Lotus Notes en Microsoft Exchange. SAP heeft ook een applicatie messaging omgeving ontwikkeld die ze Application Linking and Embedding (ALE) noemen

Voordelen:

- Men kan gedistribueerde systemen los en asynchroon koppelen;
- Men kan applicaties integreren waarvan de beschikbaarheid niet gegarandeerd hoeft te worden;
- Men kan de 'publish and subscribe' aanpak kiezen, waarbij de zendende applicatie niet hoeft te weten wie het bericht allemaal ontvangt.

Nadelen

- Applicaties moeten hun message interface gebruiken en weten wanneer en waar de rijtjes uitgelezen dienen te worden. Dit betekent dat oude applicaties aangepast dienen te worden.
- Extra inspanningen zijn noodzakelijk om het systeem 'real time' gedrag te laten simuleren.

4.3.3 Data Access/File Transfer

Een voorbeeld voor integratie door 'data access' (toegang) zijn lees en schrijf operaties op een willekeurige database. Een extern Call Centre van een onderneming zou bijvoorbeeld van de klantinformatie gebruik willen maken. De applicatie die door het call centre gebruikt wordt zou toegang tot de klantendatabase kunnen krijgen. Deze database wordt dan door berekeningen of batch routines met de centrale database gerepliceerd.

Een eenvoudig voorbeeld voor file transfer is het gebruik van ftp (File Transfer Protocol) pakketten, waarmee medewerker A een stuk data op het netwerk kan plaatsen en dit door medewerker B weer binnen gehaald kan worden.

Voorbeelden van deze techniek zijn:

- *File transfers* en *Batchloads*.
- Directe lees en schrijfrechten op externe databases, door gebruik te maken van bijvoorbeeld Open database connectivity (ODBC) koppelingen.

Voordelen:

- Handig voor het transport van grote stukken data
- Men kan handig off-line grote stukken data analyseren
- Het is recht toe recht aan en makkelijk te implementeren
- Verlangt geen veranderingen van bestaande applicaties

Nadelen:

- Lage integriteit,
- gerepliceerde data is altijd out of date,
- bedrijfslogica en regels worden omzeild,
- omkapselt niet de fysieke implementatie; dat houdt in dat alle applicaties veranderd moeten worden als er een nieuwe geïntegreerd wordt,
- Data heeft interpretatie (bijvoorbeeld conversie) nodig om informatie te worden,
- Integreert niet de bedrijfslogica van bestaande applicaties.

4.4 Conversie en transformatie

De genoemde technieken zijn bekend en uitgebreid onderzocht. In een gegeven situatie kiest men voor de ene oplossing dan weer voor de andere. De tijd speelt hierbij ook een belangrijk factor in. Oude applicaties zullen geen API hebben noch CORBA compatibel zijn.

Het echte integratieprobleem ontstaat dan ook bij het koppelen van incompatibele instanties van de boven genoemde mechanismen. Ook is het vaak zo dat men gebruik dient te maken van een uitgebreide mix van deze mechanismen, zeker in het geval van het integreren van black boxen, componenten of oude applicaties waar geen uitgaande stroom direct aan een andere kan koppelen. Deze conversies treden op door incompatibele:

- Interface technologieën

Bijvoorbeeld: conversie van IBM CICS naar Microsoft COM, of COM naar CORBA.

- Transport Technologieën

Incompatibele messaging producten en netwerk protocollen.

- Data Typen en structuren

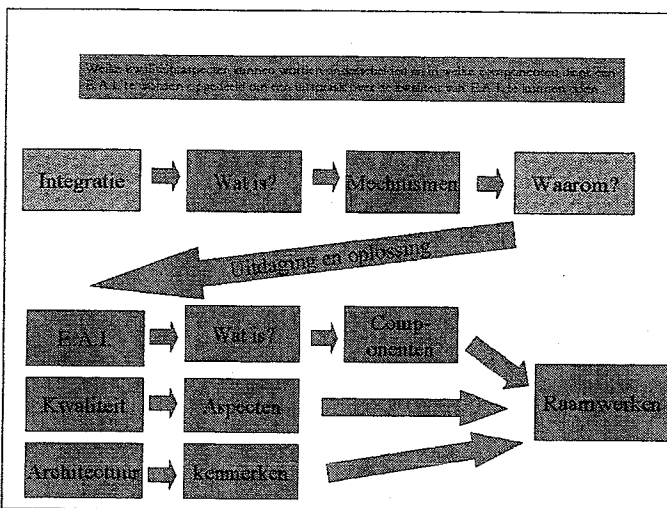
Conversie van datatypen is dagelijkse praktijk. Complexer is het formatteren en herstructureren van incompatibele data structureren in interfaces of databases.

- Integratie Mechanismen

Applicatie A zet bijvoorbeeld een bericht in een queue terwijl applicatie B alleen toegankelijk is via een API.

5 De ontstaansreden van EAI

5.1 Inleiding



Leeswijzer

Nu de mechanismen om te integreren beschreven zal nu de reden voor integratie worden toegelicht en waarom er behoefte is aan een coördinerende component, de EAI.

EAI wordt gebruikt in een omgeving die al ver geautomatiseerd is en bestaat uit deelsystemen die ook weer uit deelsystemen opgebouwd kunnen zijn.

Om een inzicht te geven in hoe het ontwikkelen van applicaties voor deelproblemen leidt tot een probleem, dat bekend staat als eiland automatisering en dat weer leidt tot de vraag naar integratie, wordt vervolgens duidelijk gemaakt naar aanleiding van de ontwikkeling van de ERP pakketten zoals we ze kennen.

Daarna zal in worden gegaan op het feit dat er een behoefte kan ontstaan aan een schakel met de buitenwereld zoals die door e-business systemen geboden wordt. Waarna duidelijk zal worden dat als men wil meespelen op die markt een soepel geïntegreerd systeem onontbeerlijk is.

5.2 De ontwikkeling van ERP pakketten

ERP pakketten (*Enterprise Resource Planning*) worden gedefinieerd door Silver, Pyke en Peterson [REF 2] als een organisatiebreed software pakket dat een onderneming in staat stelt zijn resources te beheren en alle mogelijk bedrijfsfuncties (Productie, distributie, service, financiële administratie en project management) te coördineren. Deze ERP software worden binnen steeds meer ondernemingen gebruikt.

Voor ERP pakketten werd al een basis gelegd in de jaren '60. Er werd een systeem ontwikkeld dat de materiaalbehoefte van ondernemingen stuurde, het zogenaamde, *Material Resource Planner* (MRP I). Dit systeem bestaat uit een rekenvel waarin men rekeninghoudend met levertijden en seriegrootten de bestelpunten zo bepaald dat de voorraad minimaal blijft en aan de vraag voldaan kan worden.

Door de technologische vooruitgang in de jaren '80 werden er steeds meer computers gebruikt voor dit soort berekeningen en men kwam tot de conclusie dat naast de materiaal behoefte ook andere resources zoals capaciteit van mensen en de beschikbaarheid van liquide middelen essentieel waren in de productie beheersing. Het oude systeem, MRPI, werd aangevuld met capaciteits- en planningsfunctionaliteit en zodoende onstond *Manufacturing Resource Planning* (MRP II).

Tot op dat moment werden voor deze aspecten aparte oplossingen gebruikt en vaak ook aparte software systemen. Hierdoor ontstond een effect dat bekend staat als 'eiland automatisering', oftewel het toepassen van afzonderlijke *'standalone'*-software voor specifieke bedrijfssituaties.

Toen in de jaren '90 de ontwikkelingen binnen de communicatie- en informatietechnologieën steeds verder gingen kon men door toegenomen capaciteit van de computernetwerken de systemen verder uitbreiden.

Hierbij moet gedacht worden aan bijvoorbeeld personeelsbeheer, distributie planning, onderhoudsmanagement, marketing- en kwaliteitmanagement. Deze nieuwe systemen waren ook in staat 'multi plant'operaties te ondersteunen. Dit betekent onder andere dat de productie planning over verschillende vestigingen heen verdeeld kan worden. Bovendien is het mogelijk om op verschillende locaties in verschillende landen toch een standaard manier van werken na te streven. De naam

'Manufacturing Resource Planning' wordt vanwege het integrale karakter van de nieuwe systemen vervangen door de term 'Enterprise Resource Planning' (ERP).

ERP pakketten bestaan uit meerdere, maar op elkaar afgestemde deelsystemen die elk gericht zijn op een bepaald functioneel gebied. De eerste ERP pakketten hadden minder modules dan de huidige. De meeste bedrijven hebben ook specifieke applicaties die niet te generaliseren zijn, bijvoorbeeld een stuurprogramma voor een speciaal voor één bedrijf ontwikkelde machine, of een speciaal ontwikkelde salaris module. Ook deze maatwerkoplossingen zouden in staat moeten zijn om met andere systemen te communiceren. De vraag naar compleet geïntegreerde systeem groeit naarmate de deelsystemen verder ontwikkeld werden.

5.3 De ontwikkeling van e-business

Naast ERP heeft ook het elektronisch handelen sterk aan populariteit gewonnen. Dit komt omdat bedrijven die inspelen op deze ontwikkeling in staat zijn hun klanten beter en directer te bedienen en nieuwe wereldwijde markten kunnen aanboren.

Het geheel van zakelijke handelingen die op elektronische wijze worden uitgevoerd ter verbetering van de efficiëntie en de effectiviteit van bedrijfsprocessen wordt E-business genoemd.

Waar in de jaren '70 nog veel transacties via 'Electronic Data Interchange' (EDI) protocollen verliepen is vandaag de dag het Internet de belangrijkste schakel voor transacties met de buitenwereld en neemt XML de rol van EDI over. "De markt groeit zeer sterk", zo voorspelt Forrester, "dat de totale verkopen online van 22,1 miljard Dollar in 1999 zal groeien naar 220,0 Miljard dollar in 2003, dat is goed voor 7,8 % van de totale verkopen wereldwijd". Volgens Forrester zijn er zelfs marktsegmenten die in de loop van de tijd volledig geautomatiseerd zullen zijn. Een voorbeeld is het zakelijk reizen dat in 2003 voor 31,9% online zou gaan gebeuren.

Binnen PwC wordt een model gehanteerd om de ontwikkeling van e-business aan te geven; het 4 box model. In dit model geeft elke box een bepaalde fase aan waarin een bedrijf zich in de groei naar volwassenheid kan bevinden.

Deze fasen zijn in afbeelding 3 te zien en worden hieronder beschreven.

De eerste fase (*Presence*) die bedrijven bij de ontwikkeling van hun e-business doorlopen betreft vaak louter de aanwezigheid op het net. Dit betekent dit meestal een bedrijfspresentatie en wat productinformatie. Maar ook het beschikbaar stellen van databases op het net via bij voorbeeld active server pages (.asp). In het B2B-segment (business-to-business) zetten bedrijven vaak de eerste schreden op het gebied van elektronisch gerealiseerde communicatie tussen partners en leveranciers. Momenteel bevinden zich het merendeel van de bedrijven zich nog in deze fase. Zo blijkt uit een steekproef van 38 Noord Brabantse bedrijven dat 68 % zich in deze box bevindt.

De tweede fase, *Integration*, staat in het teken van integratie en ketenmanagement, ook wel supply chain management, tussen klant en leverancier. De tweede fase kenmerkt zich doordat de informatie-uitwisseling via EDI uitgroeit tot volledige integratie met de bedrijfssystemen van zowel leveranciers als afnemers.

EDI is een standaard formaat voor het uitwisselen van bedrijfsgegevens.

Een EDI bericht bevat een string van data elementen die elk een eenduidig feit bevatten zoals prijs, product nummer en zo voorts opgedeeld door scheidingstekens.

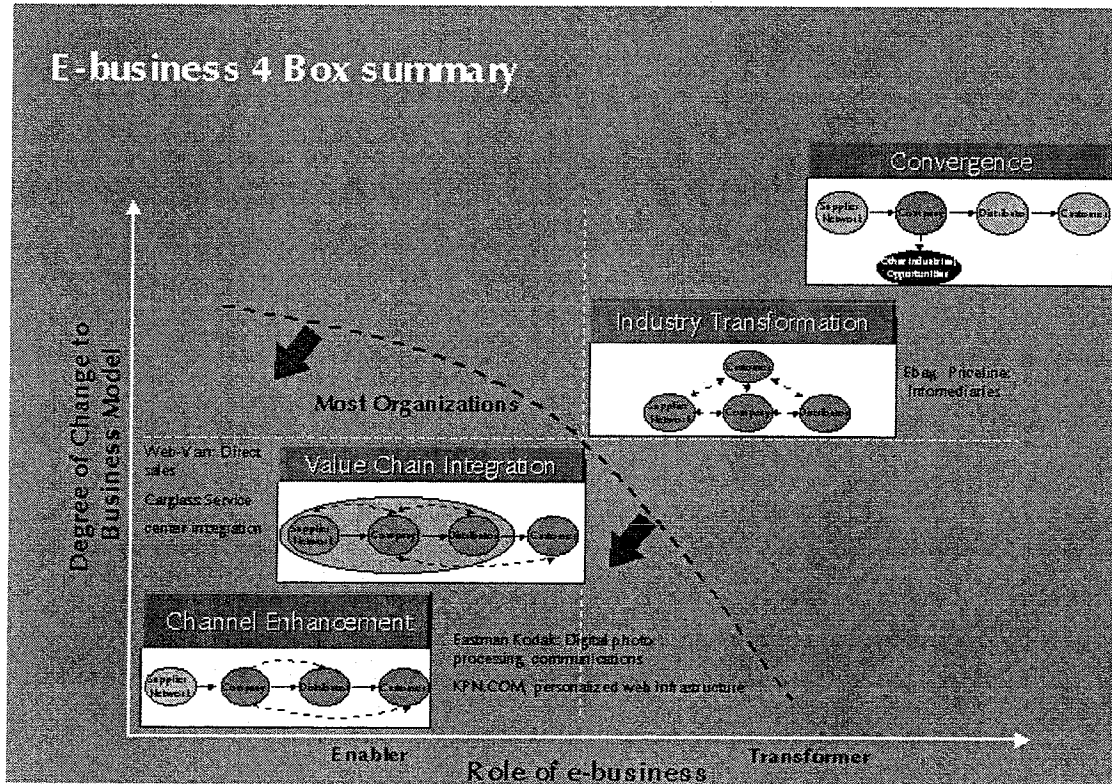
Het gebruik van *'web enabled'* ERP-toepassingen, software voor ketenbeheer en voor relatiebeheer zijn kenmerkend voor deze fase. Een typerende eigenschap van sites die zich in deze fase bevinden is bijvoorbeeld de mogelijkheid te 'track en traceren' dit betekent dat men bijvoorbeeld de status van een bestelling kan bekijken.

Vraagstukken op fiscaal en juridisch gebied worden steeds belangrijker. Dat geldt ook voor onderwerpen als risicobeheer en interne controle, aangezien bedrijfstransacties vaak via Internet lopen. Bedrijven die zich in deze fase bevinden, beginnen de problemen van 'eiland automatisering' sterker te voelen en zullen dan ook bewust worden van de behoefte aan integratie. Men kan in deze fase veel problemen oplossen door een *'web enabled'* ERP pakket te gebruiken maar dat lost alleen het probleem van het koppelen van enkele deelsystemen op. Een oplossing met een EAI dringt zich op, gezien het feit dat men zich noch verder wilt ontwikkelen en er dan meer applicaties gekoppeld moeten worden.

De derde fase, *Transformation*, in de ontwikkeling van e-business staat in het teken van de transformatie. E-business is voor het bedrijf in termen van productontwikkeling, marketing, verkoop en logistiek zo'n dominante factor geworden, dat het management genoodzaakt wordt een nieuwe visie te ontwikkelen op de kernactiviteiten van de organisatie. E-business stelt het management in staat om activiteiten te ontbundelen, zich volledig te richten op de kernactiviteiten en zo de niet-kernactiviteiten uit te besteden. Bedrijven zijn dan inmiddels op weg in de richting de vierde fase, *'Convergence'*, namelijk die van de convergentie.

Bedrijven die zich in die vierde fase bevinden, zijn vergaand geïntegreerd met andere ondernemingen, zowel binnen als buiten de eigen industrietak. Die integratie zal in korte tijd resulteren in logistieke ketens

die in dienst staan van netwerkende organisaties en markten, die de traditionele bedrijfsstructuren vervangen. In deze nieuwe dynamische netwerken staat relatiebeheer centraal. Daardoor wordt de klant meer gemak, meer keuzevrijheid en dus een forse tijdsbesparing geboden, terwijl de bedrijven profiteren van alle voordelen die bedrijfstakoverstijgende waardeketens hebben te bieden. Het is duidelijk dat ondernemingen met aspiraties op E-business gebied een goede geïntegreerde informatie omgeving nodig hebben om de beloften aan hun klanten waar te kunnen maken, qua levertijd, en juistheid. Gezien de hoge concurrentiedruk en het gemak waarmee klanten van aanbieder kunnen wisselen (men hoeft maar een andere URL in te tikken) dwingt bedrijven die in de E-commerce actief zijn altijd scherp te zijn. In de volgende paragraaf wordt een overzicht gegeven van de overige drijfveren voor een bedrijf om te integreren.

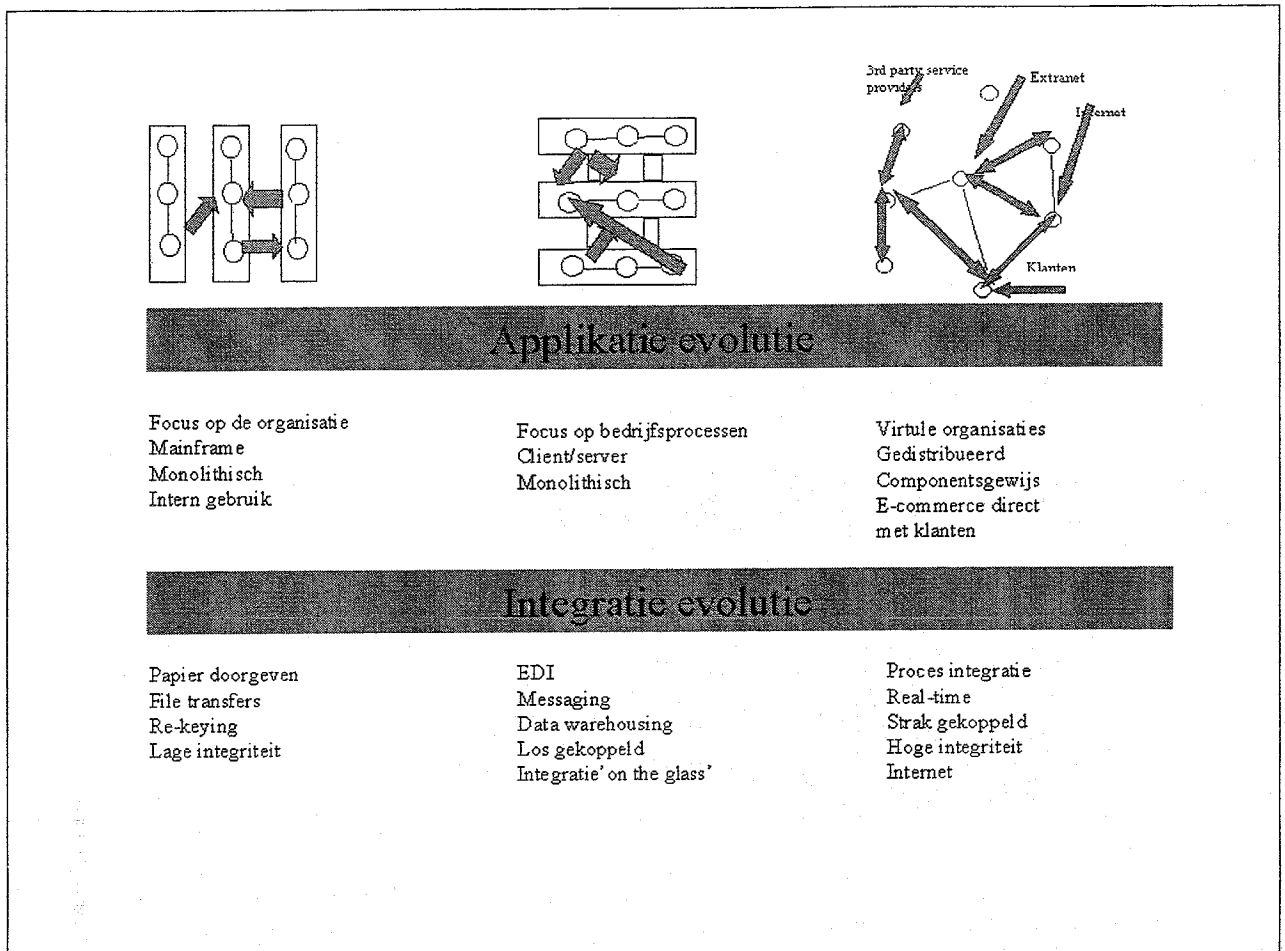


Afbeelding 7

5.4 Overige Applicaties

Het systeem landschap is door de jaren heen sterk veranderd, naast de bovengenoemde ERP en E-commerce toepassingen zijn er nog zaken die aan verandering onderhevig zijn. In deze paragraaf wordt eerst de ontwikkeling van de applicaties in het algemeen beschreven vervolgens worden de applicaties die geïntegreerd dienen te worden in groepen ingedeeld.

Tot de jaren negentig werden applicaties ontwikkeld om zelfstandig te functioneren; er waren nauwelijks koppelmogelijkheden. Deze applicaties werden ook wel kachelpijpen (stovepipes) genoemd. De integratie binnen de applicaties was strak maar tussen de applicaties los; een gevolg daarvan was dat de integriteit van de data te wensen over liet. Tegenwoordig en in de toekomst zullen de applicaties steeds meer op vormloze vlekken (blobs) lijken. Hierbij zullen delen van het systeem door partners, leveranciers en derden gecontroleerd worden en moet er dus een situatie geschapen worden waar men via interfaces en andere oplossingen tot snelle en strakke integratie komt. Een overzicht van de ontwikkeling van applicatie en integratie is te vinden in afbeelding 8.



Afbeelding 8

Organisaties hebben zelden een algemene aanpak om systemen te implementeren. Bronnen die geïntegreerd dienen te worden worden meestal alleenstaand ontworpen en worden gedraaid in verschillende omgevingen, sommige worden zelf ontwikkeld, anderen door derden.

Ontwikkelaars die nieuwe applicaties maken hebben te maken met een grootte range van programmeertalen, interfaces, protocollen en technologieën waarmee de te ontwerpen software geïntegreerd dient te worden. Om een inzicht te geven in welke applicaties er zo al zijn, en die dus samengevoegd dienen te worden, is hier onder een lijst te vinden.

De beschikbare software kan op de volgende manier ingedeeld worden:

Bedrijfssystemen: Bedrijfsbrede systemen; aangeschaft bij softwarebedrijven en ingericht voor gebruik binnen de back office van de organisatie. ERP, personeelspakketten en financiële systemen zijn de kern van veel bedrijven. Voorbeelden zijn SAP, Baan, JD Edwards en Oracle.

Front office: Bedrijfsbrede systemen aangeschaft bij softwarebedrijven met als doel het automatiseren van de processen in de front office van de organisatie. Hierbij moet worden gedacht aan Customer relationship Management pakketten of 'Sales Automation Software' (SFA). Voorbeelden zijn: Siebel, clarify, Goldmine, Vantive

Legacy applicaties: Applicaties die geërfd werden van talen, platformen en technieken ouder dan de huidige.

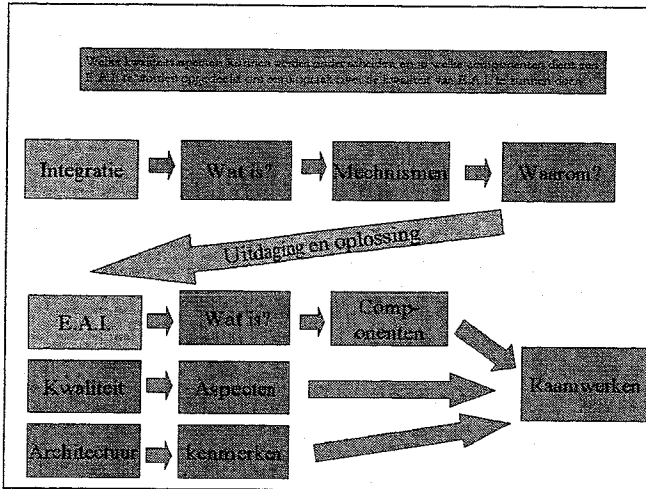
Component applicaties: Een stuk software dat een bepaalde functie vervult, herbruikbaar is en in samenwerking met andere componenten een applicatie kunnen vormen

Services: Informatie bronnen of processen die door een derde partij beheerd worden.

Middleware: Een laag waarover twee applicaties met elkaar communiceren, dit is een breed begrip volgende toepassingen vallen eronder: Netwerken, 'Object Request Brokers', 'event brokers', 'application servers', 'web servers' en 'TP monitors'

Databases: De meeste organisaties maken gebruik van een of meestal meerdere databases. Databases zijn er in allerlei soorten en typen zoals: Relationale, hiërarchische, object georiënteerde, hybride of flat file.

5.5 Drijfveren voor E.A.I



Leeswijzer

De vraag naar het samenvoegen en synchroniseren van gegevens bestaat al geruime tijd, naast oplossingen zoals 'data warehousing' (een technologie waar data uit verschillende bronnen opgehaald kan worden) groeit de vraag naar systemen die niet enkel bevroegd en gerepliceerd kunnen worden maar ook real-time eigenschappen hebben en de integriteit, dus juistheid, volledigheid en tijdigheid, kunnen garanderen. Met 'real time' bedoelen we in deze context dat de responsen van het systeem tred houden met het dominante externe proces.

Als men bijvoorbeeld online aandelen wilt verhandelen is het essentieel dat de koers op het moment waar een klant een bestelling plaatst correct is, het competitief voordeel is voor die ondernemingen die deze bestelling meteen verwerken en ze niet 's nachts moeten repliceren.

'Supply chain optimisation' (het optimaliseren van de goederenstroom) en 'call centers' zijn twee voorbeelden van bedrijfsactiviteiten waarbij het real-time bij elkaar brengen en verwerken van gegevens essentieel is. In tabel 1 staan een aantal drijfveren voor integratie opgesomd. Sommige van deze problemen kunnen opgelost worden door nieuwe systemen te plaatsen, maar met in het achterhoofd het belang van de 'time-to-market' en het resulterende competitieve voordeel en competitieve respons, betekend dat bedrijven niet de luxe hebben om de informatie systemen volledig te vervangen.

Zelfs als dat wel zou lukken komt er geen einde aan de golf van overnames en samenwerkingen die dan resulteren in het koppelen van de respectievelijke informatie systemen. Daarnaast besteden steeds meer bedrijven bepaalde taken uit, om informatie over deze processen te vergaren zal er een real-time koppeling moeten zijn.

Met de term 'disintermediatie' wordt het verwijderen van tussenpersonen in de supply chain bedoeld, dit wordt als een van de grootste voordelen van het bedrijven van e-commerce gezien. In de realiteit is het meestal zo dat de tussenpersonen niet verwijderd worden maar vervangen door anderen, dit wordt re-intermediatie genoemd, bijvoorbeeld het vervangen van winkels als verkooppunt door een bezorgdienst die er voor zorgt dat de goederen bij de klant komen.

Door het regelmatig herbeoordelen en veranderen van de supply chain ontstaat er ook behoefte aan makkelijk integreerbare oplossingen.

Business Drivers
Supply chain optimisation
E-commerce en web front end
Call centres en klant vriendelijkheid

Klant en werknemer self service
Adaptieve eigenschappen
1 op 1 marketing strategieen
Verbeteren van bedrijfsprocessen en verkorten bedrijscycli
Overnames en samenwerkingsverbanden
Virtuele organisaties en het outsourcen van diensten
Disintermediatie en re-intermediatie
Real time toegang tot informatie en integriteit van de data

Tabel 1

Het integreren van applicaties en processen is een van de nieuwe prioriteiten voor veel organisaties. De meeste applicaties zijn niet ontwikkeld om eenvoudig geïntegreerd te worden. Pas sinds kort zijn er software producenten die hun applicaties inrichten voor integratie door middel van interfaces. Maar nog steeds moet men zich aanpassen aan een bepaalde architectuur die meestal niet overeen komt met de architectuur van de andere applicaties.

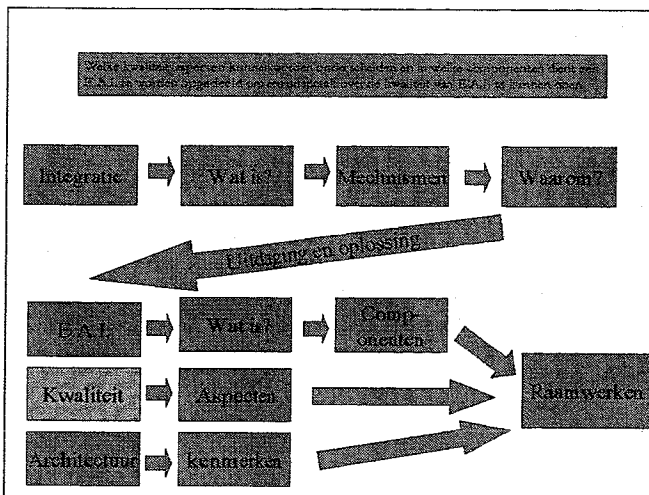
Het is onder andere het gebrek aan overeenkomstige bedrijfs- en technische architecturen die het integreren complex maken. Het is niet alleen het verschil in data formaten en interfaces, maar het gebrek aan een gemeenschappelijke definitie van de bedrijfsconcepten dat ten gronde ligt aan de verschillende bronnen die geïntegreerd dienen te worden. Het afbeelden van een 6 digit ASCII(American Standard Code for Information Interchange) getal op een 10 digit EBCDIC (extended binary-coded decimal interchange code) is relatief eenvoudig. Het afbeelden van klant informatie die het concept van meerdere locaties in een bron bevat op een ander systeem dat dat niet doet is veel complexer. Het updaten van gegevens voor een verkooporder in verschillende bronnen met totaal verschillende semantische definities van wat een verkooporder is, wordt steeds meer een nachtmerrie. Toch komt deze situatie vaak voor. Automatiseringsafdelingen hebben dus veel moeite met om incompatibele bronnen in de vorm van pakketten, platformen en legacy systemen te combineren. De uitdagingen die door deze afdeling opgelost dienen te worden zijn in tabel 2 te vinden

Uitdagingen voor de automatiseringsafdeling
Pakketten bieden nooit een complete oplossing en zijn inflexibel
Legacy moet geïntegreerd worden
Nieuwe functies moeten worden toegevoegd zijn , te duur om te vervangen en er is geen tijd om deze opnieuw te ontwikkelen
Gebrek aan een eenduidige semantiek voor zowel bedrijf als technologie
Gebrek aan interfaces
Complexe technologie met heterogene platformen, gedistribueerde systemen met n-lagen en het web
Incompatibele bedrijfssemantiek
De noodzaak om tactische en lange termijn beslissingen te balanceren

Tabel 2

6 Kwaliteit

6.1 Inleiding



Leeswijzer

Aangezien dit onderzoek bedoeld is voor de audit afdeling van PricewaterhouseCoopers zal er naar aanleiding van een geschrift van Norea [REF 3], de beroeps organisatie van IT-auditoren, een definitie van kwaliteit afgeleid worden. Met het begrip audit wordt de oordeelsvorming en rapportage over een systeem bedoeld.

Norea onderscheidt verschillende domeinen en binnen deze domeinen verschillende objecten die beoordeeld kunnen worden. Dit wordt gedaan aan de hand van kwaliteitsaspecten, de invalshoek waarover met betrekking tot een object een oordeel wordt uitgesproken, die in te delen zijn in zeven groepen. In dit hoofdstuk worden eerst de domeinen beschreven en vervolgens wordt op de kwaliteitsaspecten ingegaan.

6.2 Domeinen van IT auditing

Om beter inzicht te geven in wat er ge-audit wordt zijn de aandachtgebieden van een IT-audit ingedeeld in domeinen. Alhoewel er altijd sprake is van het auditten van een systeem wordt hiermee niet altijd het zelfde bedoeld. Een systeem kan een informatie systeem, een technisch systeem of een processysteem zijn. Systemen worden ontwikkeld, gebruikt en beheerd. Het ontwikkelen, het gebruiken en beheeren wordt gezien als processen, derhalve kan worden gesproken over het ontwikkelproces, het gebruiksproces en het beheerproces.

Onder het besturen van een proces wordt verstaan [NOREA geschrift 1]:

- Het bepalen van de doelstellingen en normen van het proces;
- De structurering van het proces;
- De planning en de aansturing van het proces;
- Het beheersen van het proces;
- Interne en externe rapportering, inclusief het afleggen van verantwoording in relatie tot de voorgaande punten;

In de praktijk is een van de moeilijkheden bij het auditten het komen tot een duidelijk en rationele opdracht. De volgende indeling in domeinen en objecten kan als hulpmiddel dienen om te komen tot een duidelijke en rationele opdracht. Andere hulpmiddelen zijn het herkennen van de levensfase van een domein of object en de aard van de processen die met betrekking tot een domein of object worden uitgevoerd.

Er worden 5 levensfasen onderscheiden:

- Het vooronderzoek;
- Het ontstaan van het systeem;
- Het in productie nemen van het systeem;

- Het operationele systeem;
- De vervanging van het systeem;

Daarnaast onderscheidt men tot de aard van processen met betrekking tot een domein:

- Het beleidsproces;
- Het ontwikkelen/ het onderhouden;
- Het gereed maken voor gebruik en beheer;
- Het gebruik;
- Het beheer;

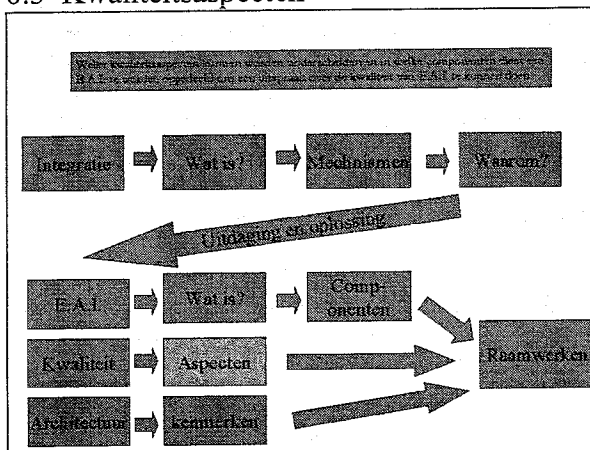
Hieronder volgt een opsomming van de domeinen van systemen zoals ze door NOREA zijn voorgesteld:

- Informatiestrategie;
- Informatiemanagement en informatietechnologie management;
- Informatiesystemen;
- Technische systemen;
- Processystemen;
- Operationele ondersteuning;

Elk van deze domeinen heeft een aantal objecten die bij zo'n domein thuishoren maar ook enkele die daar niet in thuis horen, zo kan bijvoorbeeld binnen het domein processysteem het object configuratie management beoordeeld worden, maar mag men in het domein informatiestrategie geen objecten zoals automatiseringsbeleid en automatiseringsplan onderscheiden omdat deze thuis horen bij het domein Informatiemanagement en informatietechnologie management. In deze context zijn de domeinen informatiesystemen, technische systemen en processystemen relevant aangezien een uitspraak over het functioneren van software gedaan dient te worden en niet over het gebruik of de planning van een informatiesysteem. We kiezen dus voor het gebruik van het operationele systeem.

Voor een uitgebreide beschrijving en een opsomming van de relevante objecten per domein verwijs ik naar appendix A.

6.3 Kwaliteitsaspecten



Leeswijzer

Nu de onderwerpen van de it-audit afgebakend zijn kunnen elk van de objecten op zijn kwaliteit ten aanzien van een of meerdere aspecten beoordeeld worden. Kwaliteitsaspecten worden dan ook omschreven als [REF 3]: de invalshoeken of eigenschappen waarover met betrekking tot een object een oordeel wordt uitgesproken. Kwaliteitsaspecten worden meestal uitgewerkt in een kwaliteitsboom waarbij de hoog gedefinieerde begrippen in steeds hogere detaillering naar beneden worden uitgewerkt. In dit onderzoek volstaat het de kwaliteitseisen op het hoogste niveau te beschrijven.

Men onderscheidt de volgende kwaliteitsaspecten:

- Effectiviteit;
- Efficiëntie;
- Exclusiviteit;
- Integriteit;
- Contoleerbaarheid;
- Continuïteit;
- Beheersbaarheid.

De begrippen worden in Appendix B gedefinieerd en verdeeld in deelaspecten om ze beter te kunnen duiden.

Kwaliteitsaspecten krijgen pas betekenis indien deze ten aanzien van een te beoordelen object worden beschouwd. Er zijn matrices gemaakt om voor ieder domein een beeld te geven van de samenhang tussen objecten en kwaliteitsaspecten.

Voor de domeinen van de verschillende systeemtypen worden telkens twee matrices gegeven; één voor 'ontwikkeling en onderhoud' en één voor 'het gebruik'. Dit is gedaan omdat het uitvoeren van een audit van een systeem in ontwikkeling leidt tot andere objecten van de audit en andere verwachtingen ten aanzien van de kwaliteitsaspecten waarover een uitspraak wordt gedaan, dan voor een operationeel systeem. Zo zal het aspect integriteit anders beoordeeld worden bij een systeem in ontwikkeling (alleen beoordeling opzet) dan in een werkend systeem.

6.4 kwaliteit en EAI

Maar wat is kwaliteit nu echt?

Er zijn, in de loop van de jaren, heel veel definities van het begrip 'kwaliteit' gegeven. Een greep daaruit:

- Fitness for use (geschiktheid voor gebruik) - Joseph Juran [REF 4]
- Conformance to requirements (voldoen aan specificaties) - Phil Crosby [REF 5]
- I know it when I see it
- Geheel van kenmerken van een entiteit dat betrekking heeft op het vermogen van die entiteit om kenbaar gemaakte en vanzelfsprekende behoeften te bevredigen - NEN EN ISO 8402 [REF 6]

Garvin [REF 7] onderscheidt vijf verschillende invalshoeken voor kwaliteit en komt dan ook tot vijf definities:

- Transcendente benadering: Kwaliteit is zowel in absolute als in universele zin herkenbaar. Het begrip staat voor onbuigzame normen en een hoge mate van succes;
- Productgerichte benadering: Kwaliteit is een nauwkeurig meetbare variabele. Kwaliteitsverschillen duiden op verschillen in de kwantiteit van een bepaald ingrediënt of een eigenschap van een product;
- Gebruikersgerichte benadering: Men gaat er van uit dat individuele consumenten verschillende wensen en behoeften hebben en dat zij die goederen die het best voldoen aan hun wensen beschouwen als het beste product;
- Productgerichte benadering: Kwaliteit wordt door de producent gedefinieerd als 'conform de eisen'. Wanneer een ontwerp of een specificatie eenmaal vastgesteld is, impliceert iedere afwijking een kwaliteitsvermindering;
- Prijs-kwaliteitsbenadering: een kwaliteitsproduct is een product dat een prestatie levert tegen een redelijke prijs

Al die definities bevatten een kern van waarheid, maar elk heeft ook z'n beperkingen in het gebruik. Een definitie van kwaliteit die in veel gevallen bruikbaar is gebleken (maar uiteindelijk natuurlijk dezelfde beperkingen heeft als de bovengenoemde) is deze:

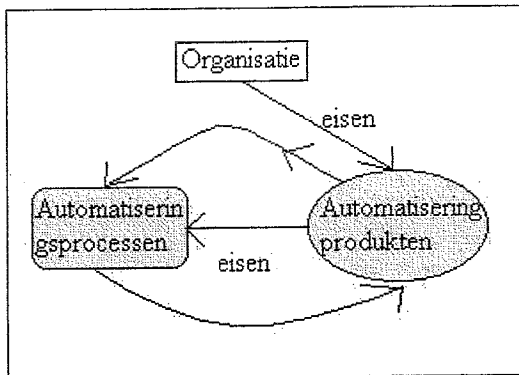
Voldoen aan de verwachtingen van de klant'

Softwarekwaliteit kan niet gelijkgesteld worden met een foutvrije oplevering van die software, in tegenstelling tot wat sommige ontwikkelaars nog steeds denken [REF 8].

In een geautomatiseerde omgeving zal de klant het management zijn en de verwachtingen de eisen die aan de software gesteld worden. Kwaliteit van de automatisering wordt dan ook gedefinieerd als:

De mate waarin het geheel van eigenschappen van een product, proces of dienst voldoet aan de eraan gestelde eisen, welke voortvloeien uit het gebruiksdoel'.

Binnen een bedrijf zullen bepaalde processen geautomatiseerd worden met automatiseringsproducten. Er is dus een duidelijke relatie tussen de automatiseringproducten en de automatiseringsprocessen. Als resultaat van de automatiseringsprocessen kan de kwaliteit van de automatiseringsproducten door de automatiseringsprocessen worden beïnvloed. Via de automatiseringsprocessen vindt daarom de beheersing van de kwaliteit van de automatisering plaats.



Afbeelding 9

Het vraagstuk ten aanzien van de kwaliteit van de automatisering bestaat dan ook uit een zodanige beheersing van de automatiseringsprocessen dat de eigenschappen van de automatiseringsproducten overeenkomen met de eisen van de organisatie. Deze eisen zullen zeker in het geval van EAI voortvloeien uit de doelstellingen van de organisatie.

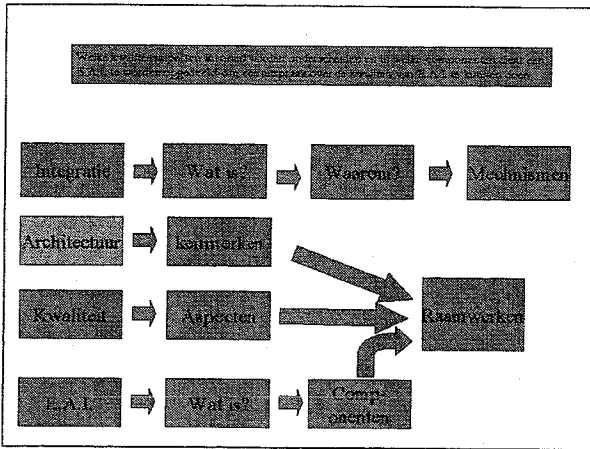
Het blijkt dat kwaliteit een abstract begrip is dat door het definiëren van domeinen en objecten en later ook normen, discreet gemaakt kan worden.

We zullen voor EAI een aantal relevante objecten moeten definiëren en voor deze dan een aantal normwaarden vaststellen om tot een zinvolle uitspraak te kunnen komen over de kwaliteit van zo een oplossing. Deze objecten met normwaarden vormen de kwaliteitseisen die we aan EAI willen stellen. In hoofdstuk 10 wordt hier uitgebreid op ingegaan en zullen ook enkel bestaande oplossingen getest worden.

Het is ook duidelijk dat een oordeel over de kwaliteit altijd implementatie afhankelijk is. De eisen, dus de doelstellingen, van een organisatie zijn verschillend. Zo zal het object real time voor een online bank een andere invulling krijgen dan voor een autoproducent.

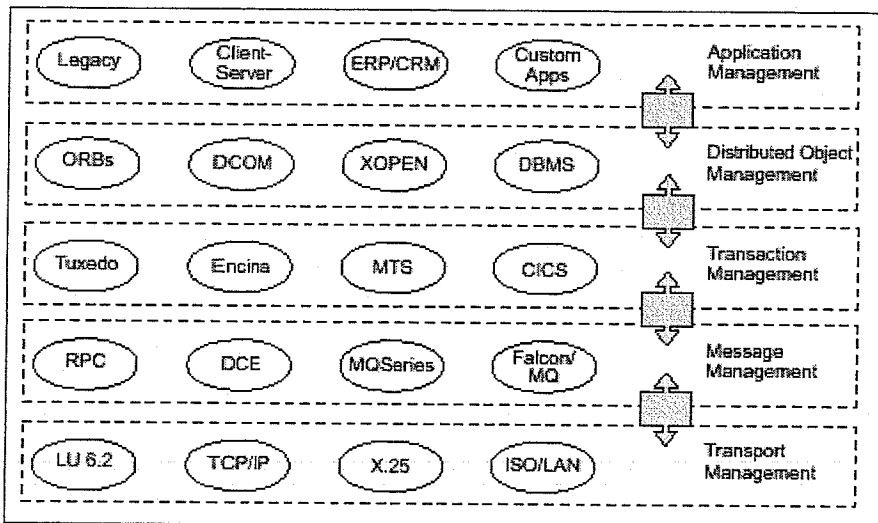
7 Architectuur

7.1 Inleiding



Leeswijzer

Aangezien een geïntegreerde omgeving met een applicatie integratie tool een complex geheel is wordt het in tweeën gedeeld. We onderscheiden de EAI component die in het volgende hoofdstuk toegelicht wordt en de onderliggende systemen. Om een indruk te geven van de verzameling software die in de verschillende lagen van een informatiesysteem beschikbaar zijn is hieronder een fictieve omgeving afgebeeld. In een omgeving met een EAI wordt boven dit smeetsel een proces laag geïntroduceerd.



Afbeelding 10

Voor het goed functioneren van een geïntegreerde omgeving is de onderliggende architectuur van groot belang. In dit hoofdstuk zal er eerst uiteengezet worden welke verschillende architecturen beschreven dienen te worden en welke hulpmiddelen er zijn zoals referentiemodellen en standaarden om een dergelijke omgeving te kunnen typeren.

Bij het ontwerpen van een architectuur zijn meerdere aspecten van belang zoals schaalbaarheid, robuustheid, betrouwbaarheid en beschikbaarheid; deze worden in de derde paragraaf beschreven. Naderhand kunnen deze kenmerken gebruikt worden om een uitspraak te kunnen doen over de kwaliteit van een breed geïntegreerd informatiesysteem door ze te mappen met de eerder besproken kwaliteitsaspecten. De eigenschappen en modellen die hier geïntroduceerd worden zijn geïnspireerd door de literatuur over gedistribueerde systemen en de beschikbare architecturen.

7.2 Het ontwerpen van een architectuur van een geïntegreerd systeem

Zonder architectuur is er geen garantie dat het ontwerp en de resulterende integratie voldoen aan de eisen. De architectuur is een wegenkaart die voordat er op detailniveau beslissingen genomen kunnen worden. Gedistribueerde systemen zijn complex en onvoorspelbaar; er moet dus zorgvuldig nagedacht worden in de ontwerpfasen. De gekozen technieken moeten bij de architectuur passen en deze ondersteunen en niet andersom. Maar de technische architectuur is maar een deel van de bedrijfsbrede architectuur; zo dienen ook de informatie architectuur en de business architectuur ontworpen te worden.

Om een integratieoplossing te maken moeten meerdere architecturen ontwikkeld worden om verschillende delen van de oplossing te beschrijven. Dit zijn onder andere de netwerkarchitectuur, de applicatiearchitectuur, de fysieke architectuur, de logische architectuur.

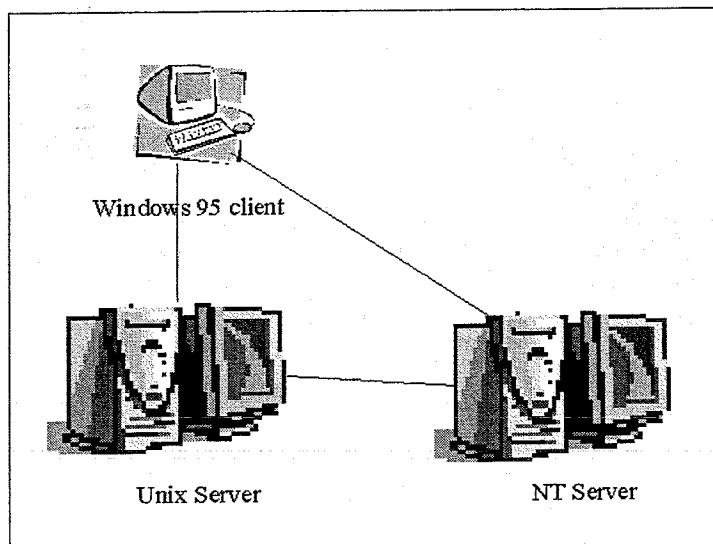
Een formele definitie van een software systeem architectuur die gangbaar is, wordt gegeven door Boehm[REF 9]:

Software Architecture = {Elements, Forms, Rationale/constraints}

Bij het ontwerpen van een architectuur moeten we de verschillende *stakeholders* (belanghebbenden) definiëren en proberen een architectuur te ontwerpen die zoveel mogelijk aan de eisen van de belanghebbenden voldoet.

7.2.1 De verschillende typen architectuur

De basis-architectuur typen zijn de logische en de fysieke. De fysieke architectuur laat zien hoe de verschillende hardware en netwerken verdeeld zijn over de gedistribueerde omgeving. De fysieke architectuur moet laten zien welke machines servers zijn welke clients en welke gebruikers er achter zitten. Een voorbeeld van een fysieke architectuur is te zien in afbeelding 10.



Afbeelding 10

Bij het maken van de fysieke architectuur van een applicatie wordt beschreven welke componenten zich waar bevinden of waar de databases, die het programma gebruikt, gestationeerd zijn.

De logische architectuur laat zien welke functies waar vervuld worden. De '*deployment mapping*' is een afbeelding van de logische op de fysieke architectuur. De deployment mapping laat de configuratie van de componenten over de fysieke systemen zien.

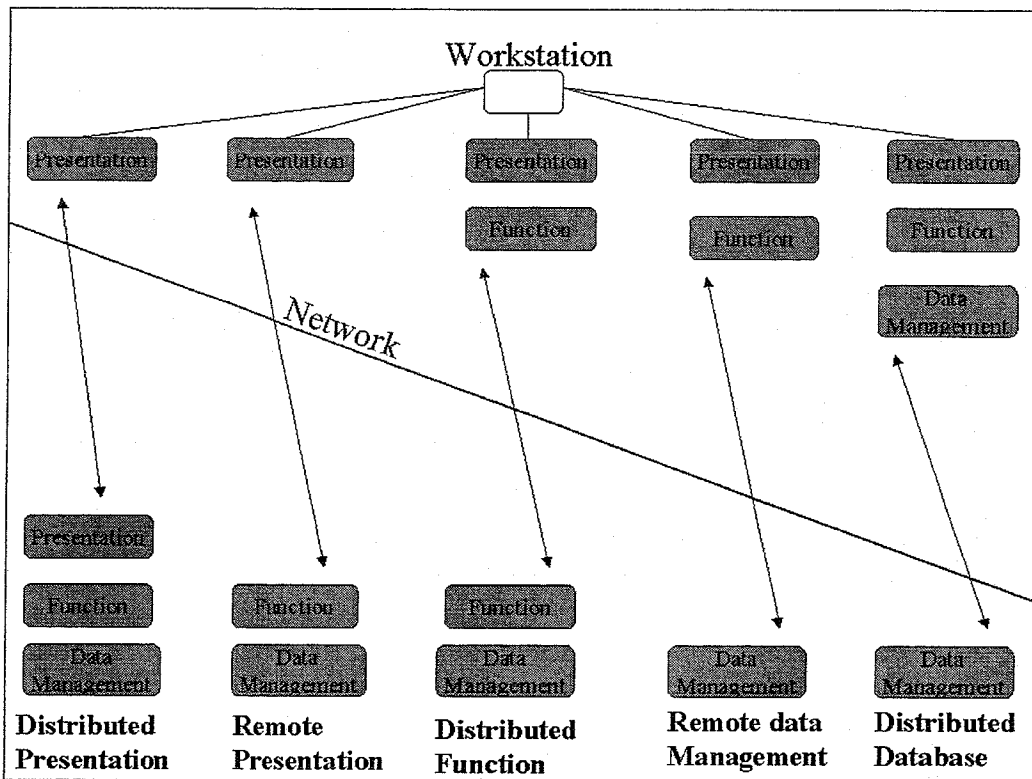
In het geval van een applicatie wordt de logische architectuur vaak beschreven met een lagenmodel. Een voorbeeld hiervan is het referentie model voor een EAI zoals in Afbeelding 15 te zien is. In het geval van een netwerk zou de logische architectuur de communicatielagen van een bepaald netwerk kunnen tonen, zoals bijvoorbeeld het OSI model.

Om een oplossing te ontwerpen moeten meerdere architecturen in zowel de logische alsook de technische view gemaakt worden. Het maken van de netwerkarchitectuur valt dan bijvoorbeeld uiteen in het maken

van een fysieke netwerk architectuur dus; de typen systemen die verbonden zijn, waar deze zich bevinden, welke communicatie lijnen er gebruikt worden en hoe ze in relatie staan met elkaar, en de logische netwerk architectuur. De logische netwerk architectuur deelt de netwerk componenten en hun configuraties op in lagen. Zo zou men kunnen zien of een componenten een bridge of router is en welke protocollen er gebruikt worden.

Naast de architecturen voor applicaties en netwerken dient er ook een zogenaamde integratie-architectuur ontworpen te worden. De integratie architectuur moet duidelijk de grenzen tussen de verschillende componenten, de externe interacties, de data conversies en de transformaties laten zien. De integratie architectuur bepaalt of een object gedistribueerd of lokaal is en hoe het samenwerkt met andere objecten.

Een belangrijke keuze die gemaakt zal moeten worden is die over wat gedistribueerd te verzorgen en wat centraal. De Gartner groep [REF 10] heeft een indeling gemaakt die loopt van 'dunne' naar 'dikke' client, deze indeling is te zien in afbeelding 11.



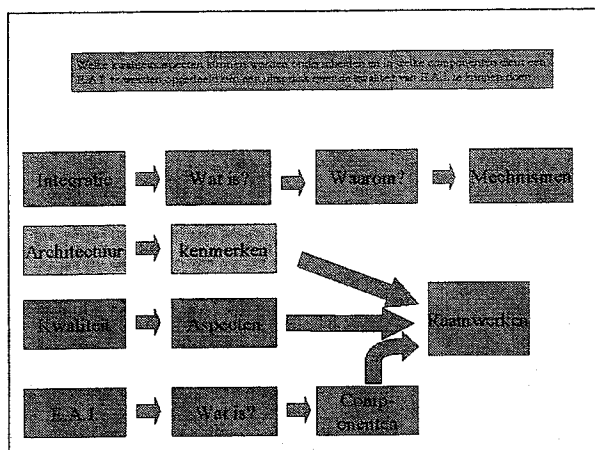
Afbeelding 11

7.2.2 Raamwerken

Er zijn verschillende raamwerken voor gedistribueerde architecturen beschikbaar. Deze raamwerken accentueren bepaalde architectuurprincipes en bevatten de definities van de kern van de communicatie-infrastructuur alsook de verzameling diensten [12]. De keuze voor een bepaald raamwerk leidt tot verdere keuzes voor bepaalde architecturen en componenten. Hieronder staat een overzicht van een aantal architectuur raamwerken, deze worden in Appendix C nader toelicht.

- Object Management Architecture (OMA)
- Windows Distibuted interNet Architecture (DNA)
- Reference Model for Open Distributed Processing (RM-ODP)
- Technical Architecture Framework for Information Management (TAFIM)
- The Open Group Architectural Framework (TOGAF)

7.3 Architectuurkenmerken



Leeswijzer

Bij het ontwerpen van een architectuur zijn er veel aspecten waar rekening mee gehouden moet worden. Er is wisselwerking tussen sommige eisen. Hiermee wordt bedoeld dat als men meer aandacht aan de ene eis geeft dan de andere, deze automatisch meer ingevuld is, zoals bijvoorbeeld bij veiligheid en performance. Door meer veiligheid controles in te bouwen wordt er meer processortijd en bandbreedte aan veiligheid ter beschikking gesteld, dit betekent dat andere berekeningen minder resources hebben en dat daardoor de performance achteruit kan gaan. Deze eisen kunnen gebruikt worden als kenmerken van een bestaand systeem, en dus ter beoordeling van de kwaliteit indien de kenmerken in relatie staan met bepaalde kwaliteitscriteria. Het is duidelijk dat gezien de verandering van technieken deze lijst regelmatig aangepast dient te worden. Men kan deze ontwerpregels zelf veel gedetailleerder omschrijven aangezien het allemaal interessante studie objecten zijn. Bij het ontwerpen van een geïntegreerde omgeving kunnen deze eisen als leidraad worden gebruikt. Het is echter te kort door de bocht om te stellen dat men hier het volledige scala dekt. Voor deze studie is het echter een praktisch bruikbare methode die redelijk snel knelpunten in de onderliggende structuur zal ontdekken.

De kenmerken zijn in acht groepen ingedeeld [REF 11]:

1. Architectonische fundering
2. Basis kenmerken
3. Geavanceerde kenmerken
4. Management
5. Robuustheid
6. Performance
7. Component relaties
8. Externe factoren

7.3.1 Architectonische fundering

De concepten die in deze paragraaf besproken worden zijn de fundering van alle systeemontwerpen.

Layering en abstractie

De meeste beslissingen die genomen worden bij het ontwerpen van een gedistribueerde softwarearchitectuur zijn wisselwerkingen tussen : *Performance*, *resource-usage*, betrouwbaarheid, en de bijbehorende kosten. Eén belangrijke eis is in hoeverre het systeem in de toekomst configureerbaar of uitbreidbaar is voor andere eisen. Het idee van abstractie is van abstracte data typen en objecten verder ontwikkelt tot het maken van lagen waarbij ontwikkel lagen op meer gespecialiseerde lagen geplaatst worden om de koppeling van de applicatie code te minimaliseren.

Hergebruik

Mowbray 1997: Waar het mogelijk is, moet de architectuur de bestaande componenten hergebruiken en zo ontworpen worden dat nieuwe componenten later hergebruikt kunnen worden

Interoperability

Wat wordt er bedoeld met interoperability? Wat wordt ermee bedoeld in relatie tot de verschillende niveaus, zoals systeem-, applicatie- en op ORB niveau? Interoperability is de mate waarin een systeem of product met andere systemen samenwerkt. Zelfs als men gebruikt maakt van het internet Inter ORB protocol (IIOP), werken de naming services van de verschillende ORB's soms niet goed samen. Zelfs als applicaties de IDL interfaces goed definiëren, is de semantiek soms verschillend en wellicht worden verschillende interfaces gedefinieerd die dezelfde functie vervullen.

Er moeten dus duidelijke afspraken gemaakt worden over naming, coding en data conventies.

Raamwerken en patronen.

Patronen zijn de laatste jaren populairder geworden. Ze worden gebruikt als een manier om herhaalbare processen en methoden om problemen op te lossen, projecten aan te pakken of om software te ontwikkelen vast te leggen. Ontwerpers en projectleiders kwamen tot de conclusie dat een bepaalde aanpak steeds weer gebruikt wordt om specifieke problemen op te lossen. Deze aanpakken zijn vastgelegd in patronen die de voorwaarden, de aanpak en voorbeelden van oplossingen beschrijven.

Ook worden tegenwoordig de zogenaamde anti-patronen vastgelegd, deze beschrijven een aanpak die tot een slecht resultaat leidt.

Een ontwikkelraamwerk kan worden gezien als een instantie van een patroon of een verzameling patronen [REF 12]. Een patroon is dan ook taalonafhankelijk. Een raamwerk wordt gemaakt voor een bepaalde programmeertaal.

7.3.2 Basis kenmerken

In deze paragraaf worden eisen en kenmerken beschreven die voor elk systeem belangrijk zijn. Deze eisen zijn generiek en moeten gelden voor het geïntegreerde systeem en zijn componenten.

Beveiliging

Beveiliging in een gedistribueerde omgeving is geen recht toe recht aan verhaal. Hoe hoger de graad van interoperabiliteit wordt zo veel groter de veiligheidsrisico's die er worden gelopen. Belangrijk is dat bij het ontwerpen van een systeem veiligheid gezien moet worden als een deel van dat systeem. Een tweede is dat veiligheid elke component en elke laag betreft. In bijlage D worden enkele voorbeelden van veiligheidstechnieken beschreven.

Persistentie

Met persistentie wordt de mate waarin componenten en hun status bewaard worden bedoeld. Traditioneel werd dit gerealiseerd door gebruik te maken van hiërarchische en relationele databases. In het geval van gedistribueerde objectsystemen wordt met persistentie het vermogen om een object status af te beelden in een persistente geheugenplaats bedoeld. Zo kan men met object georiënteerde databases de informatie over de objecttoestand manipuleren en makkelijk de relaties met ander objecten onderhouden. In een ideale architectuur wordt het feit dat een object persistent is verborgen gehouden. Als dit niet het geval is moeten clients de status via een specifieke methode expliciet bij het object opvragen. De business logic moet op de hoogte zijn welke objecten persistent zijn en welke niet. Clients zijn niet op de hoogte van het verschil tussen de objecten, en de object toestanden worden automatisch beheerd en opgeslagen.

Life cycle

Met life cycle wordt het vermogen om componenten in een systeem te beheren bedoeld. Hoe worden componenten gemaakt? Hoe worden ze gekopieerd of verplaatst? Wat is het beleid ten opzichte van het verwijderen van componenten? Hoe staan componenten in relatie met elkaar.

Als men bij voorbeeld in een object georiënteerde programmeertaal werkt bestaat er een ouder – kind relatie tussen objecten en hun scheppers. Een beleid zou kunnen zijn dat bij het verwijderen van een ouder dat ook alle kinderen verwijderd worden. Men kan voorkomen dat bij het verwijderen van een object dit object gebruikt wordt door anderen door de referenties af te lopen van het desbetreffende object. Zulke lijsten zijn lastig bij te houden in een geïntegreerde gedistribueerde omgeving, het kan zelfs leiden tot performanceproblemen. Het verplaatsen van een object kan problemen veroorzaken, een architect dient te definiëren hoe referenties aan objecten geüpdate worden en hoe de pointers naar het verplaatste object wijzen. Deze definities kunnen uiteen lopen van het *invalid* verklaren van een referentie tot het ontwikkelen van mechanismen om het bewegen van objecten inzichtelijk te maken voor de gebruiker.

Naamgevingsstrategie

Componenten in dezelfde proces- of applicatieruimte gebruiken vaak pointers om te refereren aan informatie die ze nodig hebben. In een gedistribueerde omgeving kunnen pointerafhankelijkheden ontstaan en sommige systemen ondersteunen ze überhaupt niet. Dit betekent dat ze op een andere manier objecten dienen te identificeren. Het Transport Control Protocol (TCP) en het User Datagram Protocol (UDP) gebruiken bekende poorten als namen van beschikbare diensten. Het DCE en CORBA hebben beide naming diensten waar natuurlijke taal afgebeeld wordt op diensten van machines. Beslissingen die genomen moeten worden zijn : Wordt de naamruimte plat of hiërarchisch? Hoe kan er aan een component gerefereerd worden bij het integreren van componenten van derden met zelf ontwikkelde componenten? Zijn de namen zo georganiseerd dat ze makkelijk uitbreidbaar zijn? Wordt er rekening gehouden met de veiligheid, oftewel kunnen applicaties met een 'foute' naam andere componenten misbruiken.

Identificatie van objecten

Er zijn twee methoden die gebruikt worden om object te identificeren. Identificatie van objecten is belangrijk omdat clients objecten kunnen lokaliseren en dat ze een onderscheid kunnen maken tussen de verschillende instanties van objecten. Een methode is gebruik te maken van pointers. Deze methode werkt goed voor een applicatie op een locatie echter niet in een geïntegreerde gedistribueerde omgeving. In een gedistribueerde omgeving wordt gebruik gemaakt van objectreferenties om objecten te identificeren. Een ontwerper hoeft niet precies te weten hoe deze referenties opgebouwd zijn of waarvoor ze staan, meestal wordt er een combinatie de Unique User Identifier (UUID) en het netwerk en de host- informatie gebruikt om bepaalde instantie van een object te identificeren. Maar ook bij het gebruik van referenties dringen zich een aantal vragen op:

Hoe worden referenties tussen verschillende object systemen vertaald?

Wat gebeurt er met een referentie naar een object als het object crasht of verwijderd wordt?

Wat gebeurt als men een object verplaatst en er wordt een nieuwe referentie gecreëerd?

Uiteindelijk moet er een mechanisme gemaakt worden waarmee objecten geïdentificeerd worden, dit mechanisme moet een unieke identificatie hebben die zo lang als het object bestaat geldt, dat bestand is tegen crashes en verplaatsingen en persistent is.

Events

Geïntegreerde omgevingen zijn verzamelingen van verschillende applicaties die met elkaar samenwerken om een bepaalde bekwaamheid aan de eindgebruiker kunnen bieden. Zulke coöperaties moeten kunnen reageren op onverwachte gebeurtenissen. Meestal moeten na zo een gebeurtenis, bekend onder de Engelse term *event*, meerdere applicaties transacties uitvoeren. Een bekende oplossing hiervoor is het al eerder beschreven 'publish en subscribe' model. Ook java omgevingen kunnen goed met events omgaan.

7.3.3 Geavanceerde kenmerken

Metadata

Informatie over gegevens noemt men metadata. Met metadata in de context van applicatie integratie bedoelt men semantische informatie over data en interfaces die tussen de systeem elementen gedeeld worden en de informatie die nodig is om de verschillende instanties van objecten te kunnen onderscheiden. Om de metadata tussen communicerende applicaties of gebruikers te kunnen verspreiden moet de structuur en de bedoeling van de metadata bekend zijn bij alle participanten. Dit kan door een template aan te bieden voor de metadata, of een 'meta'-beschrijving over hoe de informatie gegroepeerd wordt ter beschikking te stellen. Als applicaties weten hoe er gegroepeerd werd kunnen ze beslissingen nemen door instanties van die metadata te interpreteren met de template. De ontwerper van de architectuur dient dus een metamodel, van de informatie die in de metadata komt, te maken. Verschillende niveaus van metadata zijn soms nodig in een gedistribueerd systeem. De metadata over een bepaalde interface is de syntaxis en de semantiek behorende bij deze interface. Een ander type metadata is in de vorm van informatie over componenten typen of namen en of ze ondersteund worden door de diensten aangeboden door de gedistribueerde infrastructuur zoals COM+.

Data formaat

Aangezien componenten van systemen steeds meer divers worden, hebben deze steeds vaker eigen dataformaten. Dit veroorzaakt afhankelijkheden van bepaalde versies van dataformaten en is gevoelig en moeilijk te onderhouden. In een omgeving met veel dataformaten moet de architect afwegen of het

noodzakelijk is dat alle applicaties op de hoogte moeten zijn van de dataformaten om met elkaar te communiceren. Er dient een keuze gemaakt te worden waar de data geconverteerd wordt als dat nodig blijkt. Er zijn drie mogelijkheden:

Client makes it right (CMIR);
Server makes it right (SMIR);
Middle makes it right (MMIR).

Er is een wisselwerking tussen onderhoudbaarheid en efficiëntie.

Batch of transactie georiënteerd

Oudere applicaties verplaatsen vaak groepen data in een keer, de batch verwerking, deze vindt dan vaak 's nachts plaats. De transactie georiënteerde applicaties verplaatsen veel kleine hoeveelheden data.

Internationalisering

Internationalisering is het mogelijk maken om informatie zo te laten zien dat deze voldoet aan de lokale configuraties. De verschillen kunnen zich voordoen in het verschil in formaat van dezelfde informatie (tijd, eenheden, valuta) of het verschil van het presenteren van hetzelfde bericht in verschillende talen, en culturele eisen aan GUI's.

Dataformaten moeten hetzelfde zijn over heel het net. Applicaties met user interfaces maken de data leesbaar door ze te vertalen in een string daarna *parsen* ze de informatie van de string-representatie in het algemene dataformaat. Vandaag de hebben zowel Unix als Microsoft besturingssystemen die instelbare scan en formaterings eigenschappen hebben.

Veel internationaliseringsraamwerken bieden de gebruiker uit te breiden verzamelingen van berichten geïndexeerd naar taal en getal, waardoor berichten in vele talen kunnen worden getoond. Kwesties waar gelet op moet worden is de locatie en distributie van deze berichtscatalogi. Men kan ze centraal op de server opslaan of men download de client met de bijbehorende berichten op de lokale machine.

7.3.4 Management

Om een geïntegreerd systeem te gebruiken moet er een standaard beleid zijn om nieuwe software of een update te installeren, een EAI component kan hier veel hulp bij bieden. Deze managed als het ware het gedistribueerde systeem.

Men moet het systeem kunnen bekijken en veranderingen in de status kunnen constateren. Het management van een geïntegreerde omgeving houdt het draaiend krijgen en houden van het gehele systeem in. Men onderscheidt de hieronder beschreven aandachtspunten

Onderhoudbaarheid

De software moet zo ontworpen zijn dat men problemen snel herkent en men ze makkelijk kan herstellen. Als meerdere componenten een fout veroorzaken is het soms lastig de oorzaak te vinden en vast te stellen bij welke component de fout begonnen is.

Versie beheer

De beheerder van een systeem moet op de hoogte zijn van veranderingen in de interface die veroorzaakt worden door nieuwe systeemcomponenten alsook voor functionele veranderingen met betrekking tot de functionaliteiten van componenten. Het versie beheer zou er op gericht moeten zijn dat het systeem *backwards compatible* is. Dat men ten alle tijden een stapje terug kan nemen naar een werkende versie. Een aanpak om dit te realiseren is om aan objecten die onveranderbaar zijn een unieke *identifier* toe te kennen. Deze zou equivalent kunnen zijn aan de naam van een klasse of een bepaalde interface. Ook moet er een mechanisme gemaakt worden waarbij componenten de versie van andere componenten kunnen constateren. Verder moeten er verschillende versies van de integratie software beschikbaar zijn omdat het upgraden vaak gefaseerd gebeurt.

Testbaarheid en het opsporen van fouten

Foutopsporen ook wel *debugging* is een wetenschap die zich nog aan het ontwikkelen is. Er zijn tools ontwikkeld die in een geïntegreerde omgeving de ontwikkelaar de mogelijkheid bieden de stroom van uitvoeringen tussen de verschillende componenten in kaart te brengen. De meeste tools bieden deze mogelijkheid echter nog niet. De ontwikkelaar moet dan aan de hand van de debugs van de alleenstaande componenten zelf een beeld vormen van de communicatie stroom. Dit is voor het ontwikkelen van een systeem nog wel te accepteren; is men echter in de operationele fase dan is dit niet voldoende. Componenten moeten zo ontworpen worden dat ze de interne toestand en informatie over de omgeving opslaan in het geval van een fout. Zoals bekend bestaat een systeemlandschap vaak uit oudere applicaties die gemaakt werden om alleen te draaien en deze zijn dus niet uitgerust om zulke informatie te verstrekken. Sommige EAI componenten proberen de datastromen in kaart te brengen om daarna het probleem te isoleren. Deze methode werkt als de fout bij één applicatie optreedt, maar is er bijvoorbeeld een deadlocksituatie waar meerdere componenten een rol spelen, dan faalt de methode.

Configuratie

In een gedistribueerde omgeving is de configureerbaarheid van de individuele componenten een belangrijke factor om in het schatten hoe *'manageble'* het systeem is. In een homogene omgeving, dat wil zeggen alle machines zijn bijvoorbeeld solaris, kunnen de configuratie mechanismen van het netwerk gebruikt worden. Bij gemixte omgevingen is dit lastiger. Sommige EAI componenten bieden de mogelijkheid een abstracte laag toe te voegen en daar de instellingen te verzorgen.

7.3.5 Robuustheid

Componenten in een systeem en het hele systeem moeten aan bepaalde gebruikerseisen voldoen om het systeem succesvol in een operationele omgeving te laten functioneren. Deze eisen kunnen worden gebruikt bij het beoordelen van de kwaliteit van een systeem.

Schaalbaarheid

Schaalbaarheid is een maatstaf voor hoe goed een systeem zijn taken kan blijven vervullen als er meer te verwerken valt. Ideaal gezien zou het toevoegen van gebruikers moeten leiden tot een voorspelbare uitbreiding van de *'resources'* zoals geheugen en bandbreedte. Als de vraag echter exponentieel toeneemt hebben we een niet-schaalbaarsysteem. Gedistribueerde systemen bieden mogelijkheden voor schaalbaarheid maar er zijn ook veel gevaren. Er zijn veel factoren die in de schaalbaarheidsfunctie opgenomen dienen te worden. De ontwikkelaar heeft echter ook meer mogelijkheden om gebruik te maken van extra capaciteit die bij het toevoegen van machines ontstaat. Diensten die via het netwerk worden aangeboden moeten zo ontwikkeld worden dat de data die verstuurd wordt geminimaliseerd wordt. De interfaces moeten er voor zorgen dat het aantal oproepen via het netwerk geminimaliseerd wordt alsook het aantal betrokken componenten.

Betrouwbaarheid

Een systeem dat de kern van een bedrijf vormt, moet betrouwbaar, goed getest en in staat zijn foutmeldingen degelijk te verwerken. In een omgeving waar meerdere componenten samenwerken kan men de betrouwbaarheid meten door de gemiddelde tijd tussen fouten te bepalen, oftewel de MTBF (Mean Time Between Failures). In het geval van een enkele applicatie is de MTBF een indicatie van hoe goed de code werd geschreven. Betrouwbaarheid is afhankelijk van de zwakste component in het systeem. Een niet betrouwbare sleutelcomponent verlaagt de betrouwbaarheid van het hele systeem. Een zelden gebruikte component die nauwelijks faalt beïnvloed de betrouwbaarheid van het systeem niet. In een geïntegreerde omgeving spelen naast de applicaties alle andere componenten ook een belangrijke rol voor de betrouwbaarheid. Heeft men bijvoorbeeld een slecht netwerk dan kunnen de software componenten zo goed zijn als mogelijk is maar zal de betrouwbaarheid niet verbeteren.

Beschikbaarheid

Beschikbaarheid is een maat voor het aantal tijdseenheden waar een systeem *'up'* is om een aanvraag te verwerken vergeleken met de tijd die nodig is om zo een aanvraag te verwerken. Fouttolerantie en betrouwbaarheid helpen de beschikbaarheid van een systeem.

Fouttolerantie

Is het acceptabel dat netwerkproblemen het systeem onbruikbaar maken? Wat gebeurt er als de database server crasht? Zijn dit voldoende redenen voor het systeem om te stoppen met het aanbieden van zijn diensten? De bekwaamheid voor een systeem om te kunnen blijven functioneren na het optreden van een fout noemt men fouttolerantie.

Herstart en herstel

Het is belangrijk voor een systeem dat het zichzelf kan herstellen na een fout zonder dat het hele systeem afgesloten moet worden. Sommige EAI componenten hebben de mogelijkheid een gecrashte component op te sporen en deze dan individueel af te sluiten en te herstarten

7.3.6 Performance

Er zijn meerdere methoden om de performance van een software systeem te verhogen. Het verbeteren van de performance van een applicatie is kostbaar. Performance in een geïntegreerde omgeving is uitdagend. De individuele componenten zijn weliswaar voorspelbaar, maar het samenspel kan leiden tot een 'gevecht' voor bronnen. De bandbreedte van het onderliggende netwerk, de grootte en aantal data dat verplaatst wordt en de technologie die in de infrastructuur gebruikt worden beïnvloeden de performance.

Throughput

Throughput is een maat voor het aantal werk dat uitgevoerd kan worden per tijdseenheid. Meestal gebeurt dit met een benchmark.

Multithreading

Tegenwoordig bieden veel programmeertalen de mogelijkheid tot multithreading. Een multithreading applicatie kan meerdere 'draden' hebben. Met een draad wordt dan bedoeld een draad van controle of een draad van uitvoering. Zo kan een applicatie op een server meerdere threads hebben voor de verschillende clients, zodat elke client direct en tegelijk met de anderen geholpen kunnen worden.

Threading heeft zin voor applicaties die I/O gebonden zijn. Zulke applicaties wachten lange tijd op het gereed komen van een proces buiten hun adresruimte. Multithreading kan dus in sommige situaties de performance verhogen maar brengt altijd een verhoogt deadlock risico met zich mee.

Dispatching en Load balancing

In een gekoppeld systeem kunnen clients in sommige gevallen de server uitkiezen waar ze een bepaalde actie willen laten uitvoeren. Het is het verstandigst als de client die server kiest die het snelste zijn aanvraag kan verwerken. De *load* geeft aan hoe druk een bepaalde applicatie is. De client kiest dan de service-applicatie met de laagste *load*. Maar hoe weet een client welke dat is. Een oplossing is het toevoegen van een dispatchinglaag. Een oplossing voor de keuze is bijvoorbeeld *round robin* oftewel het sequentieel aflopen van de beschikbare mogelijkheden. Dit is niet een bijzonder intelligente oplossing maar werkt desondanks redelijk in de praktijk. Complexere dispatchdiensten kiezen de minst drukke applicatie maar houden tevens rekening met externe factoren zoals de kosten van het kiezen van een bepaalde dienst.

Caching/smart proxies

Als men het aantal aanvragen over het netwerk minimaliseert komt dat de performance ten goede, omdat het datatransport de bottleneck van een gedistribueerd systeem is. Caching is een bekende techniek om dit mee te realiseren. Als bijvoorbeeld een client sequentieel een aantal waarden uit een databasetabel opvraagt kan het verstandig zijn heel de databasetabel beschikbaar te stellen op de client omdat er een logische kans is dat er weer aanvragen aan de database gegeven worden. Clientapplicaties kunnen dit op applicatieniveau oplossen of gebruik maken van *smart proxy* mechanismen die veel aanbieders vandaag de dag hebben. Data die vaak opgevraagd wordt, kan ook op die manier beschikbaar gesteld worden. Er moet wel voor gezorgd worden dat ge-cachte informatie overeenstemt met de data op de originele datadrager.

7.3.7 Component relaties

Een van de belangrijkste eigenschappen van een geïntegreerde omgeving is het samenspel tussen de componenten. In de logische architectuur beschrijving worden deze relaties vastgelegd. In deze paragraaf beschrijven we de verschillende mogelijkheden de componenten te laten samenwerken en wat daarbij komt kijken. De meeste EAI oplossingen bieden hulp om deze aan deze eisen te voldoen.

Interface

Zoals eerder beschreven is een interface de manier om een component integreerbaar te maken. De Interface moet de diensten van de component aanbieden en het *contract* tussen client en een server die de clients aanvraag afhandelt opstellen.

User interfaces

Omdat geïntegreerde systemen steeds vaker als bedrijfssysteem ingezet worden is er veel werk verzet om een *'overall'* blik op het systeem te kunnen geven. Een mogelijkheid hiertoe biedt het dashboard van een EAI component waar het samenspel tussen de componenten gecoördineerd wordt zonder dat men details kent over hoe de componenten samenwerken.

Vererven en aggregeren

Vererven of *Inheritance* is een statisch mechanisme waarmee software specialisatie relaties gemodelleerd kunnen worden. Aggregatie is een heel-gedeelte relatie tussen software componenten. Dit kan statisch gedefinieerd worden om de zin van systeemobjecten te tonen, of ze kunnen dynamisch onderhouden worden waar aggregatie een *'bucket'* of een verzameling van gerelateerde componenten is.

Volgorde en controlemechanismen

Zoals al eerder opgemerkt is het essentieel voor een bedrijfsbreed geïntegreerd systeem dat de taken in de juiste volgorde uitgevoerd worden. Zo zal een veiligheidscheck om een client te autoriseren altijd eerst afgerond moeten zijn voor het proces verder gaat, anders falen alle andere acties omdat deze veronderstellen dat het een ongeautoriseerde client betreft.

Gedistribueerde omgevingen bieden de mogelijkheid veel threads of control te hebben; dit kan betekenen dat er met gedeelde resources gewerkt wordt, wat kan leiden tot deadlocks. Als een component bepaalde bronnen claimt is het dan ook belangrijk dat hij ze binnen eindige tijd vrij geeft.

Tijd

In een gedistribueerde omgeving is de notie van tijd lastiger dan bij een *'stand-alone'* applicatie. Er zijn twee aspecten: ten eerste hoe zorgen we er voor dat alle applicaties consistent dezelfde tijd hebben en het tweede aspect heeft te maken met de timing en daardoor de bepaling van de volgorde van de afronding van instructies.

Een centrale klok zou een oplossing kunnen zijn voor het probleem om een consistente tijdsdefinitie voor collaborerende applicatie te vinden. Maar dan blijft het probleem van het zenden van de tijd naar applicaties en clients. Hoe wordt er met de netwerk vertragingen omgegaan? Een vaakgebruikte oplossing is om de tijd op een machine dominant te laten zijn voor de applicaties op die machine. De machines zorgen er dan voor dat hun tijd in pas loopt met enkele ander machines met betrouwbare tijdsbronnen. Dit wordt geregeld met het Network Time Protocol (NTP).

7.3.8 Externe factoren

In deze paragraaf komen die punten aan bod die buiten de controle vallen van een EAI component en meestal zelfs buiten de controle van de systeemarchitect vallen. Enkele zijn technisch en andere gebaseerd op organisatiekundige of culturele aspecten.

Standaarden

Bij het ontwikkelen van een bedrijfsbreed systeem of componenten ervan moet er altijd rekening gehouden worden met standaarden in de desbetreffende industrie. De eerder beschreven raamwerken kunnen hier een goede hulp bij zijn, zoals ook de eerder besproken protocollen zoals TCP/IP

Protocollen

Een protocol is een afspraak tussen twee entiteiten over hoe de communicatie zal plaatsvinden. Netwerk protocollen zoals TCP/IP laten twee systemen informatie uitwisselen via een connectie georiënteerde communicatie sessie. De architectuur moet eenvoudig bruggen tussen verschillende protocollen toelaten.

Connectivity

Connectivity behelst *protocol gateways*, maar deze nemen aan dat de hardware de communicatie ondersteund. De protocol- en performanceeisen voor een bepaald deelsysteem moeten nauwkeurig onderzocht worden. Er moet een pad gevonden worden met voldoende bandbreedte voor de aangevraagde communicatie. De keuzen lopen uiteen van het kiezen van een modem voor een inbelverbinding tot het aanschaffen van routers die het System Network Architecture (SNA) vertalen in TCP/IP en het opzetten van voldoende interne en externe bandbreedte.

Externe beperkingen

Binnen elke organisatie staan er banden met adviseurs of softwarehuizen waar het management waarde aan hecht. Dan zal men bepaalde concessies met betrekking tot de softwareleveranciers moeten accepteren. Het kan zo zijn dat partners een bepaald systeem gebruiken en men graag met hen samen wilt werken en daarom een bepaalde vorm EDI moet gebruiken.

Organisatiekundige aspecten

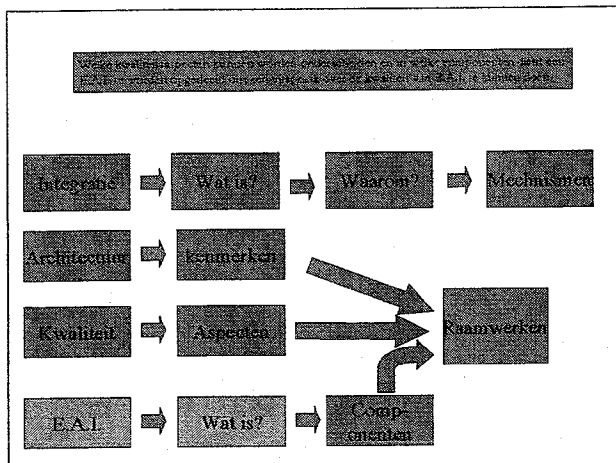
Organisatiekundige aspecten worden vaak onderschat maar ze spelen wel degelijk een belangrijke rol. Voorbeelden zijn de problemen die ontstaan als verschillende afdelingen het eens moeten worden hoe de klant benaderd dient te worden. Een ander aspect is de vraag om zelf aan te schaffen of de eigen IT afdeling het te laten maken. Er kunnen weerstanden tegen nauwere samenwerkingen. Het is ook zo dat soms afdelingen onderling concurreren en er helemaal niet zo blij mee zijn dat alle informatie gedeeld wordt. Deze weerstanden kunnen worden voorkomen door goede voorlichting.

7.4 Controle Raamwerk

Nu er een verzameling kenmerken beschikbaar is, en het begrip kwaliteit ontleed is in kwaliteitsaspecten, kan er een relatie gelegd worden tussen de kenmerken en de kwaliteitsaspecten. Hierdoor ontstaat een matrix waarin men kan zien op welke kwaliteitsaspecten een kenmerk invloed uitoefent. Het is niet zo dat een kenmerk slechts één kwaliteitsaspect beïnvloed, meestal zullen er meerdere aspecten afhankelijk zijn van een kenmerk. Om te voorkomen dat alle aspecten in relatie staan met een kenmerk is er voor gekozen telkens de drie meest relevante aspecten aan te kruisen. De resulterende matrix is te zien in appendix E, hier wordt in de meest rechtse kolom ook een motivatie gegeven waarom de gekozen aspecten het meest relevant zijn.

8 De bouwstenen van EAI

8.1 Inleiding



Leeswijzer

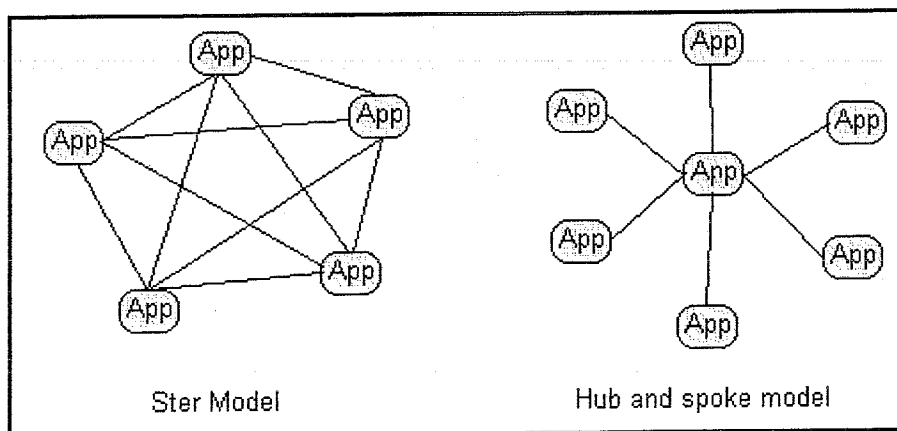
Na het in kaart brengen van de onderliggende architectuur wordt in dit hoofdstuk de EAI component besproken.

Enterprise Application Integration (EAI) beschrijft een nieuwe categorie software producten die de integratie van bestaande en nieuwe applicaties mogelijk maakt en coördineert. EAI software bevindt zich in een laag boven het hiervoor beschreven bestaande systeem. Daar ligt ook de kracht van EAI het kan bestaande koppelingen in kaart brengen en bestaande systemen koppelen met nieuwe systemen. Als een EAI component reeds in gebruik is, is het integreren van nieuwe applicaties met de bestaande infrastructuur een stuk eenvoudiger.

Een dergelijke component heeft een aantal essentiële bouwstenen waardoor het zijn functie kan vervullen. Voordat de functie nader geformuleerd wordt met behulp van een model zullen eerst drie belangrijke bouwstenen worden beschreven: de verbidingsarchitectuur, de verbindingen en het procesmodelleer gereedschap.

8.2 De Verbindingsarchitectuur

Organisaties die applicaties willen integreren kunnen gebruik maken drie mogelijke applicatie integratie schema's: het 'publish and subscribe' model, het 'ster' model, of het 'hub and spoke' model; de laatste twee zijn in afbeelding 12 afgebeeld.



Afbeelding 12

In het 'ster model' heeft elke applicatie een adaptor of connector voor elke aanwezige applicatie. Dit type interfaces komt overeen met het al besproken 'point to point' model. Voor elke set interfaces is

afzonderlijk programmeer werk noodzakelijk. Voor kleine applicatieportfolio's is dit interessant, voor grote aantallen applicaties leidt dit model tot extreem hoge kosten.

Het 'hub and spoke' model bestaat uit een centrale *Hub*, engels voor centrum, die alle spaken, de applicaties met elkaar verbindt. Applicatiefunctionaliteiten zoals messagequeueën, 'message routing', EDI, en *data-translation* worden allemaal door de hub verzorgd. Dit model is effectiever voor grotere applicatie portfolio's. Wel is specifieke software nodig om de *interfaceconnectivity* te definiëren en te managen. De Hub wordt in dit geval gebruikt als controle object. Een proces of workflow die van af de Hub wordt geïnvokeerd wordt in processtappen verdeeld en roept de desbetreffende events aan over de spaken en controleert het geheel van het proces. Het 'publish en subscribe' model werkt anders. Informatie wordt ter beschikking gesteld zonder dat de zender weet welke applicaties ervan gebruik gaan maken. Deze manier van informatie verspreiden wordt ook wel broadcasting genoemd en wordt gebruikt als er geen koppeling of zelfs respons van de andere applicatie noodzakelijk is. Een variant is de bus architectuur, waar de berichten op een logische bus gezet worden en niet door een centrale Hub moeten.

Het aantal verbindingen zo laag mogelijk houden zal er voor zorgen dat de complexiteit onder controle blijft. De 'hub en spoke' aanpak is bekend voor het beperkte aantal verbindingen, het is een techniek die zowel de complexiteit onder controle houdt en toekomstige uitbreidingen eenvoudig houdt. Dit komt omdat nieuwe applicaties alleen met de Hub verbonden dienen te worden en de functionaliteit van de Hub alleen centraal hoeft worden uitgebreid om de nieuwe applicatie in het systeem te integreren.

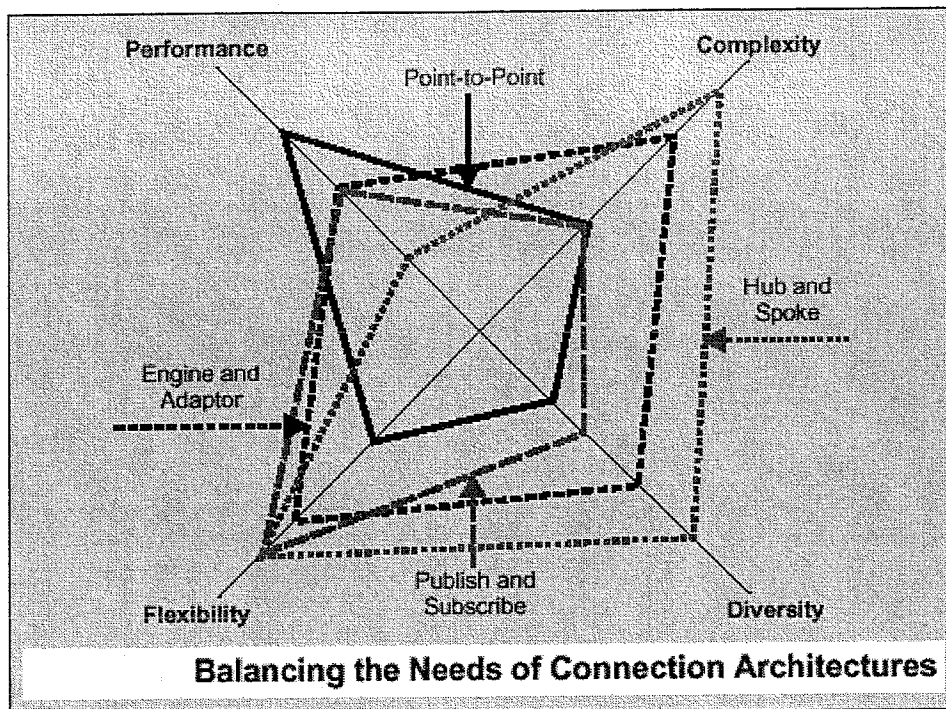
Om een verstandige beslissing te kunnen nemen over de meest preferente architectuur moet men met meerdere factoren rekening houden. 'Hub en spoke' is zeker zeer flexibel maar het kan onnodige performance overhead veroorzaken, omdat ieder bericht bekeken wordt nadat de 4GL regels geïnterpreteerd zijn verder gestuurd worden naar de zelfde applicatie als de reeks berichten daarvoor. 'Publish en subscribe' biedt een andere vorm van flexibiliteit; het is niet noodzakelijk de inschrijvende (subscribing) partij te integreren. Maar de complexiteit van de bedrijfsregels of de diversiteit van technologieën maakt 'publish en subscribe' complexer.

Om een architecturale beslissing te nemen moeten we de verschillende systemen met elkaar kunnen vergelijken. Dit is niet eenvoudig de meeste bedrijven gebruiken dan ook een mengvorm.

Een aantal factoren die een rol spelen zijn:

- *Performance*: Moeten de verbindingen ten alle tijde een hoge performance hebben?
- *Diversity*: Hoeveel verschillende applicaties en informatiebronnen dienen geïntegreerd te worden?
- *Flexibility*: Worden bedrijfsregels en integratie bronnen regelmatig gewijzigd?
- *Complexity*: Is de integratie simpel of complex?

Er bestaan modellen waarin deze modellen tegen elkaar afgewogen worden. Zo heeft de Butler group in 2000 volgende onderzoeksresultaten bekend gemaakt:



Afbeelding 13

8.3 Adaptoren en connectoren

Adaptoren bieden de mogelijkheid verschillende applicaties met elkaar of met een centraal systeem te verbinden. Connectoren bieden verschillende typen verbindingen:

Op componentniveau worden operaties tussen componenten, objecten en applicaties verzorgt. Meestal wordt dit gedaan door een broker die de onderliggende componenten verbindt. Men moet dus op de hoogte zijn van het onderliggende object model van de gesnookerde component. Bij het gebruik van dit type connectoren moet de aanwender dus van het ene data model op het andere mappen. Dit type connector is het moeilijkst te ontwikkelen en heeft een standaard om de interface te definiëren nodig. Zo moet men bijvoorbeeld bij het gebruik van CORBA de methoden van objecten definiëren in de Interface Definition Language (IDL).

Op API niveau wordt een van tevoren gedefinieerde ingang naar applicaties; applicatie moet de semantiek van de aangeroepen applicatie begrijpen.

Op technisch niveau worden de onderliggende lagen gebruikt om data door te geven. Hiermee wordt bedoeld dat men bijvoorbeeld Mqseries gebruikt om een bericht naar een applicatie te sturen of direct gebruik makend van SQL (Standard Query Language) een database te bevragen.

Connectoren geven toegang tot ingekapselde informatie bronnen. Er is een breed assortiment van informatiebronnen in een systeem zoals interfaces, databases en voorgedefinieerde transacties.

Connectoren bieden bronnen de mogelijkheid 'event driven' te werken. Een event is geformatteerd bericht dat uitgewisseld wordt tussen de bron en de vragende partij. Een event kan de trigger zijn van een bepaald gedrag en kan de respons weer opvangen. Daar komt bij dat een bron de respons zou kunnen publiceren, uitgeven, via de connector door alle subscribers van deze bron een bericht te sturen. Dit noemt men de 'publish and subscribe' methode, zoals in het vorige sub-hoofdstuk beschreven werd.

Connectoren voeren de vertaling uit van een binnenkomend event naar een type gedrag dat door de bron herkend kan worden. Ook vertalen ze de respons in een formaat dat door de vrager begrepen wordt. De categorieën die door de meeste aanbieders van EAI software geboden worden zijn de volgende:

- Database connectoren

Deze connectoren sturen events die vertaald worden naar SQL of een andere query taal die door de database begrepen wordt. De respons is het resultaat van de uitvoering van de query op de database. Een

database connector kan ook een Remote Procedure Call (RPC) op een DBMS invokeren of andere methoden van een DBMS aanroepen;

- Interface emulatoren

Deze brede categorie bevat een variëteit aan 'green screen emulatoren', 'screen scrapers' en terminal emulatoren. 'Screen scraper' zijn stukken software die het mogelijk maken voor een pc om karakter per karakter van een mainframe op te vangen, en deze in een makkelijk te gebruiken GUI representeren. Deze connectoren verstopten de complexiteit van het opvragen van data achter niet grafische schermen. Zij transformeren toetsaanslagen en beeld reacties in events;

- Enterprisesysteem connectoren

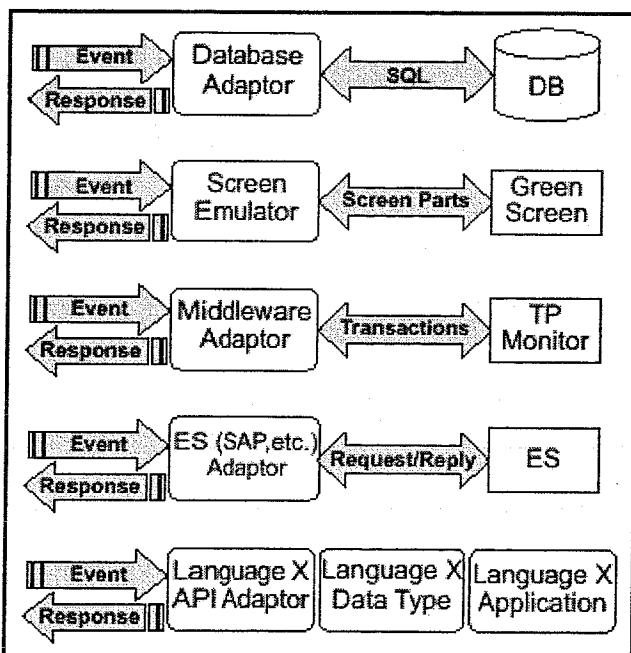
Deze connectoren verbinden de bekende Enterprise pakketten, dus CRM en ERP pakketten. De connector kent de semantiek van het doelsysteem. Zo zal een SAP connector een model genereren dat betekenis heeft voor het SAP systeem;

- Programmeertaal connectoren

Deze connectoren bieden de mogelijkheid voor de aanvrager om gebruik te maken van de taal van de doelapplicatie om procedures te invokeren of data op te vragen. Een 'C' connector herkend C datastructuren en stuurt een aanvraag naar de doelapplicatie. Bijvoorbeeld het converteren van een database veld in een HTML string;

- Eigen connectoren ontwikkelomgeving

Er bestaan geen one-size-fits-all connectoren en er zijn vele bedrijfspecifieke situaties en omgevingen. Daarom bieden veel aanbieders van EAI software ontwikkelomgevingen die het mogelijk maken eigen connectoren te definiëren. Deze 'custom connectoren' kunnen gecodeerd worden om zelfgemaakte datastructuren en eigen 'event handlers' voor interne en extern gegereerde events te verwerken. Een zelfgemaakte connector kan alleen gemaakt worden als men de semantiek van zowel bron als doelsysteem kent. Dit coderen wordt een aantal programmeertalen gedaan. Zo moet men om een connector met SAP te maken gebruik maken van BAPI's (Business Programming Application Interface), SAP's eigen standaard interface taal. Afbeelding 14 laat de verschillende connectoren nog eens zien.



Afbeelding 14

Connectoren zijn objectgeoriënteerde constructies. Een connector kapselt de software van het doelsysteem in en maakt er een component van. Een connector gedraagt zich als een interface van die component en biedt de service van de component. Zo worden oude applicaties, 'legacy'- systeemcomponenten. De publieke interface laat een verzameling services, transacties en procedures zien die geïnvokeerd kunnen worden. Er worden ook wel eens uitbereidingen aan de interface toegevoegd om de functionaliteit van het legacy systeem te verbeteren. Het is duidelijk dat de meest bekende adaptoren in elke EAI oplossing te vinden moeten zijn, of in ieder geval eenvoudig gemaakt kunnen worden. Het aantal verschillende

adaptors is een indicatie van hoeveel verschillende pakketten geïntegreerd kunnen worden en tevens een belangrijk kwaliteitsaspect.

8.4 Workflow management

Naast adaptoren en de architectuur is die software die de processen van een bedrijf in kaart brengt en waar nieuwe processen gedefinieerd kunnen worden essentieel. Voor het in kaart brengen, ontwerpen en analyseren van processen zijn softwareproducten onder de noemer workflowmanagement ontwikkeld. Daarom wordt er nu op workflow management in gegaan. De definitie van workflow management, of in het Nederlands: werkstroomautomatisering, is vastgesteld door The Workflow management Coalition als:

The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rule.

Er zijn verschillende tools op de markt die helpen bij procesdefinitie, maar ook procesmonitoren die helpen bij het in de gaten houden van processen. Er zijn echter meer taken die door workflow management gedekt dienen te worden. In Appendix F zullen deze eisen beschreven worden en zal er aan de hand van de richtlijnen van de workflow management coalition (WfMC) een raamwerk worden afgeleid.

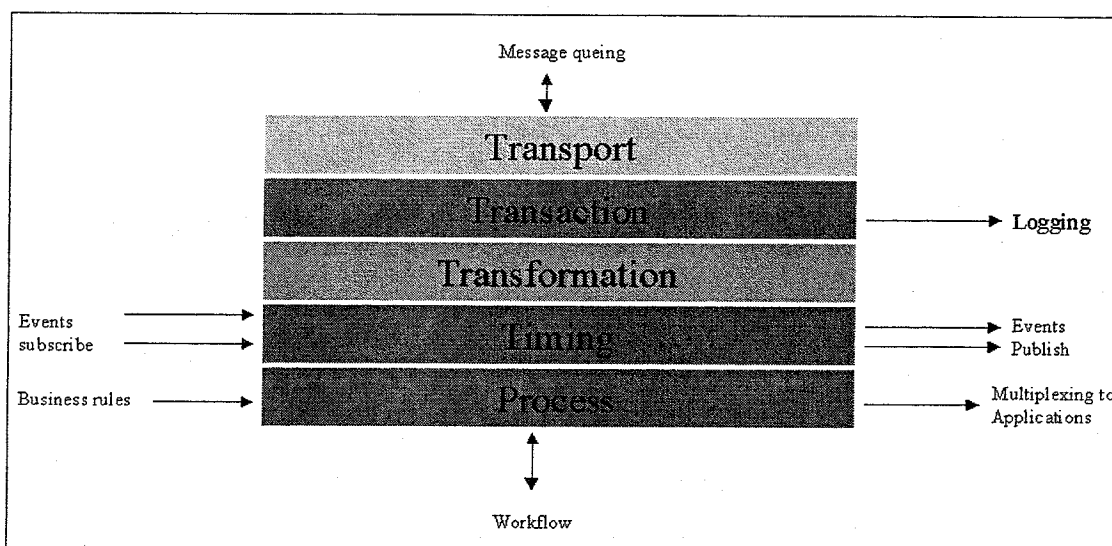
9 Een Raamwerk voor EAI

9.1 Inleiding

Nu de gebruikerseisen van een EAI bekend zijn; namelijk het mogelijk maken en het coördineren van de integratie van bestaande systemen en nieuwe systemen; wordt de volgende stap het afleiden van een model van EAI. Bij het ontwikkelen van software in deze grootte en complexiteit worden er op hoog niveau lagen modellen afgeleid. Een dergelijk model wordt ook wel de logische applicatiearchitectuur genoemd. Dit model wordt gebruikt om taken binnen een applicatie te benoemen en er een bepaalde hiërarchie in aan te brengen. Dit onderzoek richt zich echter niet op het ontwerpen van een nieuwe EAI applicatie maar op het beoordelen van de kwaliteit van een gegeven oplossing. Hiervoor is de logische applicatiearchitectuur ongeschikt omdat men moeilijk in een oplossing kan kijken zonder de bron code of de laag niveau-modellen ter beschikking te hebben. Daarom wordt er ook een dienstenmodel voor een EAI afgeleid; dit model beschrijft zoals de naam al doet vermoeden de diensten die een EAI te vervullen heeft. Met dit model kunnen oplossingen getypeerd worden door leveranciers een vragenlijst te laten beantwoorden of de beschrijving van de software te analyseren.

9.2 Referentie model voor een EAI component

In dit referentie model wordt, gebruikmakend van de beschrijving van de logische applicatiearchitectuur, aangegeven welke functionaliteiten en functionaliteitsgebieden een EAI moet hebben om een succesvol product te zijn. De verschillende lagen worden hieronder toegelicht.



Afbeelding 15

Per laag moet nu aangegeven worden welke verantwoordelijkheden deze heeft:

9.3.1 Transportlaag

Deze laag is verantwoordelijk voor het fysieke bezorgen van de informatie, of voor de verbindingen tussen de bronnen die geïntegreerd dienen te worden, en de kwaliteit van deze dienst. Taken zijn het verzorgen van:

- **Netwerk verbindingen:** Hoe en waar worden de verbindingen gerealiseerd. Hoe worden verschillende netwerk protocollen geadresseerd;
- **Berichten:** Is de informatie een bericht (message-based) en heeft het een message queuing product nodig? Moet de levering van een bericht gegarandeerd worden;

- Veiligheid en encryptie: Hoe worden de verschillende veiligheidseisen van elke applicatie of informatiebron behandeld? Is single log-on een eis? Moeten de berichten gecodeerd worden? Waar zou dit moeten gebeuren?;
- Restart en hertel: Wat gebeurt in geval van storingen?;
- Routing: Waar gaat een bericht naar toe en wie bepaald en coördineert de route van een bericht?

9.3.2 De transactie laag

Deze laag verzorgt de integriteit en het beheren van de transacties die bij applicatie integratie nodig zijn. Om real-time integriteit te kunnen garanderen, moet in deze laag ook de notie van gedistribueerde transacties verzorgd worden. Hoe bijvoorbeeld 'two-phase commit', back-out, en recovery approach afgehandeld worden.

- Transactie management: Hoe worden gedistribueerde transacties afgehandeld? Hoe wordt de integriteit van de transacties gegarandeerd? Hoe worden toestanden behandeld?;
- Fout tolerantie: Welke mechanismen worden gebruikt bij fouten in de verbinding? Hoe worden fouten opgespoord?;
- Logging en monitoring: Worden transacties gelogged? Welke analyses m.b.t. de integratie zijn noodzakelijk uit het oogpunt van frequentie, performance enz.

9.3.3 De transformatie laag

In deze laag wordt de data, die door de te integreren applicaties wordt uitgewisseld, geconverteerd. De conversie van data typen en interfaces eenvoudig is, maar het converteren van business objects is lastiger, soms is er extra informatie nodig van andere applicaties of een coördinerend proces.

- Data- en protocolconversie: Hoe worden de verschillende data typen geconverteerd? Hoe worden protocol verschillen afgehandeld?;
- Semantieke conversie: Hoe worden de verschillen in business objects afgehandeld?;
- Messageformatting: Hoe worden berichten geformatteerd? Hoe worden problemen zoals lege velden opgelost?;
- Interface conversion: Hoe worden de verschillen in technische- en bedrijfsinterfaces opgelost?;
- Datavalidatie: Wat voor mechanismen zijn er om data te valideren (type checking, content validation)?;
- Waar wordt er gevalideerd (voor het bericht gezonden wordt, bij ontvangst, enz.)?;
- Hoe wordt gecodeerde data getransformeerd?

9.3.4 Timing

In deze laag ligt de verantwoordelijkheid voor *wanneer* informatie tussen applicaties uitgewisseld moet worden. Soms triggered een event een ander maar ook een two phase commit of een asynchrone afhandeling is denkbaar. Soms is real-time noodzakelijk, soms kan 's nachts repliceren voldoen. Om dit te coördineren is een beheersend mechanisme noodzakelijk. Er zijn veel modellen voor de interactie tussen applicaties, zoals:

- Request and reply;
- Conversational (synchronous);
- Publish and Subscribe;
- Asynchronous.

De timing laag verzorgt dan ook volgende zaken:

- Scheduling: Wanneer moet informatie uitgewisseld worden?;
- Events: Welke events moeten er gemonitord worden? Welke applicaties moeten er geïnformeerd worden?;
- Interaction Model: Worden synchrone en asynchrone activiteiten ondersteund? Zijn 'publish en subscribe' technieken noodzakelijk?

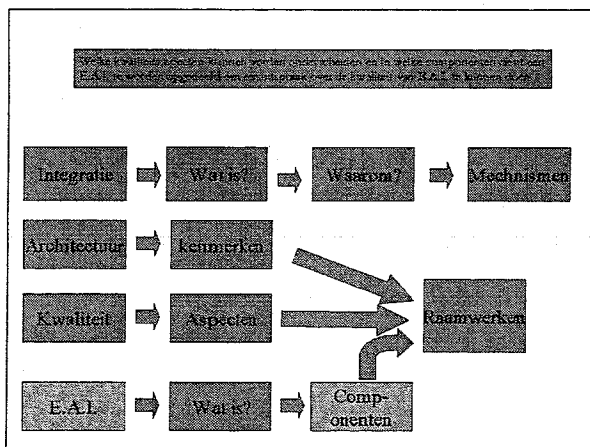
9.3.5 De proces-laag

In de proces laag worden de *business-rules* die de integratie bepalen, afgehandeld. Uiteraard hebben deze rules, regels die de bedrijfsvoering typeren, effect op meerdere lagen, bij de complexere rules moet bijvoorbeeld de content, de inhoud van een bericht, gecontroleerd worden en meerdere applicaties gecoördineerd worden. Dit is zoals al eerder opgemerkt een typisch probleem dat met behulp van een de workflow management component van een EAI oplossing opgelost kan worden.

- Multiplexing: Zijn 1 op N en N op N interacties noodzakelijk? Hoe wordt dit gemanaged?;
- Business rules: Kunnen alle business rules uitgedrukt worden? Waar worden deze gemodelleerd? Waar worden ze gecodeerd?;
- Workflow: Moet er een heel bedrijfsproces uitgevoerd om de integratie te beïnvloeden? Is de integratie zelf een bedrijfsproces? Wordt de integratie in meerdere stappen met meerdere bronnen uitgevoerd?

Dit referentiemodel is geschikt om een goed inzicht in de structuur en werkwijze van een EAI te vinden. Het is de logische architectuur van de applicatie deze is echter minder geschikt voor het doen van harde uitspraken over hoe het een en ander geregeld voor een specifieke oplossing geregeld is. Daarom wordt er nu een dienstenmodel afgeleid dat in de praktijk beter bruikbaar is.

9.4 Dienstenmodel



Leeswijzer

In dit hoofdstuk wordt een model beschreven aan de hand waarvan uitspraken zullen worden gedaan over de kwaliteit van EAI oplossingen.

Het maken van een model heeft als doel te abstraheren en wel zodanig dat algemene uitspraken gedaan kunnen worden. Een probleem hierbij is het vinden van de juiste diepgang om te voorkomen dat één bepaalde oplossing te specifiek wordt beschreven of het model te algemeen blijft om gefundeerde uitspraken te kunnen doen.

In het gekozen model worden de EAI oplossingen allereerst opgedeeld in componenten, gebaseerd op bepaalde functionele eisen. Daarna worden de componenten in relatie gebracht tot die kwaliteitsaspecten die het meest relevant worden geacht in relatie tot het controlframework.

De indeling van componenten is gebaseerd op de dienst die een component levert in het EAI. Bijvoorbeeld een component die de dataoverdracht verzorgt. De diensten in een EAI zijn afgeleid van de bouwstenen en het idee achter applicatie integratie dat verwoord werd in het referentie model. Deze komen overeen met de eisen die belanghebbenden (*stakeholders*) aan een integratie oplossing stellen en die ook door de onderzochte aanbieders op een of andere manier vervuld worden.

Per EAI leverancier (oplossing) wordt vervolgens in kaart gebracht hoe deze diensten zijn ingevuld. Hierbij wordt opgemerkt dat niet alle EAI oplossingen alle diensten bieden. Sommige aanbieders specialiseren zich op bepaalde diensten en maken voor de overige gebruik van oplossingen van derden.

Op basis van de mapping tussen diensten en kwaliteitsaspecten ontstaat een matrix op grond waarvan de onderlinge relaties tussen kwaliteitsaspecten en de diensten in kaart worden gebracht.

Deze matrix zal inzicht verschaffen in de relatieve relevantie van diensten op de onderzochte kwaliteitsaspecten.

Hierbij wordt opgemerkt dat het veelal niet mogelijk zal zijn absolute uitspraken te doen. Afhankelijk van de gekozen invalshoek zal een dienst meestal wel in verband kunnen worden gebracht met een zeker kwaliteitsaspect (N:N relatie).

De gekozen benadering is dan ook uitsluitend gericht op om in hoofdlijnen inzicht te verschaffen ten aanzien van de meest belangrijke relaties tussen diensten en kwaliteitsaspecten.

Op deze wijze zal het model in elk geval inzicht kunnen geven in de belangrijkste afbreukrisico's als een bepaalde dienst niet aan een relevant kwaliteitsaspect voldoet.

Op basis van het model zullen vervolgens eigenschappen (Zie Hoofdstuk De EAI bouwstenen) worden afgeleid. Deze zullen op een bepaalde manier gegroepeerd dienen te worden. Ook hier zal weer de juiste balans inzake diepgang gevonden moeten worden.

Voorbeeld: Om een proces te kunnen modelleren zou de EAI over een dienst moeten beschikken die een dergelijke modelering mogelijk maakt.

De EAI zou dus over een Grafische Interface moeten beschikken. De vraag die hierbij opkomt is of een dergelijke Grafische Interface als aparte dienst dient te worden beschouwd of dat een dergelijke dienst voor de beschrijving van het model buiten beschouwing gelaten kunnen worden?

Dit keuzevraagstuk is van toepassing op meerdere veelal kleine componenten waaruit een EAI kan zijn opgebouwd. Een te gedetailleerde benadering zou kunnen leiden tot een in de praktijk niet goed handelbaar model.

De enige oplossing hierin is het maken van keuzen. Evenwel zullen deze altijd arbitrair zijn. Hierop wordt in het navolgende ingegaan.

9.4.1 Het ontwerp van het model

In deze paragraaf wordt toegelicht hoe het model tot stand gekomen is en waarom bepaalde keuzen zijn gemaakt.

Hierbij wordt uit gegaan van een Top Down-benadering waarbij vanuit de hoogste naar de laagste semantische integratie graad wordt afgedaald (Zie hoofdstuk 4)

De start in de benadering ligt dan ook in de integratie op procesniveau. Achtereenvolgens zullen de diverse diensten op hoofdlijnen worden toegelicht.

Procesniveau

- Op proces niveau kunnen twee verschillende processen worden onderscheiden:
- Een intern proces:

Een proces waarbij uitsluitend applicaties, al dan niet netwerkoverschrijdend, van het eigen bedrijf gebruikt worden. Over deze applicaties kunnen duidelijke afspraken worden gemaakt, bijvoorbeeld over de wijze waarop ze zo soepel mogelijke geïntegreerd kunnen worden. Veiligheidsrisico's treden vooral op als de applicaties op verschillende locaties draaien en dus via een netwerk wordt gecommuniceerd.

- Extern proces

Het tweede type proces is een proces waarbij met een partner of klant gecommuniceerd dient te worden. Zulke processen worden ook wel met een *buzzword* B2B genoemd. Een voorbeeld is het automatisch bestellen van goederen door een ERP pakket zodra een bepaald voorraadsniveau wordt bereikt. Hier komen veel meer aspecten om de hoek kijken. In hoeverre is een bedrijf bereid om zijn eigen applicaties open te stellen voor businesspartners. In hoeverre passen de klanten zich aan of in hoeverre is het systeem al hiervoor geschikt. Het zal duidelijk zijn dat bij bedrijfsoverschrijdende processen de veiligheidsaspecten zwaar wegen. Verder zullen de partners geadministreerd dienen te worden en zal een analyse en registratie van transacties dienen plaats te vinden.

We zullen de dienst voor het beheren van interne processen *Business process automation* noemen en het integreren van applicaties met applicaties van derden *Inter-Enterprise Automation*.

Naast de proceslaag moet ook een dienst beschikbaar zijn die de processen tot uitvoering brengt en de mogelijkheid voor applicaties om met elkaar te communiceren coördineert.

Verder zullen er mechanismen moeten zijn die de processen in stappen opdelen en deze vervolgens toekennen aan applicaties die deze uitvoeren. Deze regels worden meestal *business rules* genoemd.

De mogelijkheden om deze regels succesvol te implementeren worden in hoofdstuk 4 besproken.

Voorbeelden hiervan zijn events en 'content-based' routing.

Ook hier worden twee niveaus onderscheiden. Het hogere niveau zorgt voor de vertaling van de business rules en het herkennen van events, zodanig dat de processen uitgevoerd kunnen worden en dat na het succesvol beëindigen van een taak van het proces de volgende opgeroepen wordt. Deze dienst noemen we *Integration Logic*. Het lagere niveau, met de naam *content transformation*, zorgt ervoor dat de applicaties in hun eigen taal worden aangesproken. Hier worden bijvoorbeeld de events vertaald naar het juiste formaat, of de data in de juiste semantiek getransformeerd.

Als de data of objecten tot een verwerkbare vorm zijn aangepast moeten ze naar de applicaties gezonden worden. Deze dienst wordt in eerste instantie in twee stukken opgedeeld. Er zal een mechanisme nodig zijn om de data fysiek te verplaatsen en er zal een mechanisme nodig zijn dat de interactie met de applicatie verzorgt.

Voor beide diensten is gekozen deze te splitsen. Voor de communicatiedienst wordt onderscheiden:

- de software die het data transport coördineert zoals bijvoorbeeld een message queuing systeem, *de communicational services*
- het onderliggende netwerk, *de infrastructural foundation*.

Het onderscheid is aangebracht omdat de verschillende EAI op verschillende platformen met verschillende bericht systemen geïkt zijn, en men daar dus een duidelijk verschil kan herkennen.

Verder is het communicatie gedeelte typisch een dienst die ook door derden vervuld kan worden.

Zo zijn er EAI's die gebruik maken van een NT netwerk en MQ-series. Deze keuzen typeren de EAI en beperken zijn toepassingsgebied.

De interactie dienst wordt ook in twee componenten opgesplitst. Ten eerste de adaptoren (*Adaptors and connectors*) zoals beschreven in het hoofdstuk 8.3 connectoren en adaptoren. Deze component wordt afzonderlijk bekeken omdat hierin grote verschillen zitten. Zo zal een oplossing met een uitgebreid scala aan adaptoren meer integratie mogelijkheden bieden dan een oplossing met een beperkt aantal adaptoren. Verder is het type adaptor van belang. Complexe adaptoren zoals bijvoorbeeld voor een ERP pakket dienen standaard geleverd te worden omdat het lastig is deze later zelf te maken. Een belangrijke reden hiervoor is dat de klanten vaak niet de beschikking hebben over het complete datamodel van het ERP pakket.

Een andere verschil kan hem zitten in de eenvoud van het maken van nieuwe adaptoren. Zo zijn er aanbieders die een gebruikersvriendelijke grafische omgeving ter beschikking stellen en zijn er anderen waar nieuwe adaptoren 'met de hand' in C geschreven moeten worden.

Een adaptor moet, nadat een applicatie een bericht stuurt, dit bericht aanbieden aan de EAI. Hiervoor moet er dus een bepaalde communicatie vorm afgesproken worden. Men kan kiezen uit meerdere

communicatie modellen zoals 'Publish/Subscribe', 'Request/Respon's' of 'Data push', deze dienst wordt *Application Interaction* genoemd. De werkingwijze van deze technieken werd beschreven in hoofdstuk 8.

Op basis van voornoemde lagen kan de communicatie nu in principe op alle integratie niveaus plaatsvinden.

Er is echter voor gekozen nog twee diensten op te nemen omdat deze in alle oplossingen voorkomen en een belangrijk deel van de oplossing vormen. Deze diensten zijn niet zo zeer van belang voor het primaire proces van EAI maar meer voor de ondersteuning.

De eerste dienst zorgt voor een intern consistent data model. In dit model wordt na de transformatie de informatie opgeslagen. We noemen deze dienst *Metadata services*.

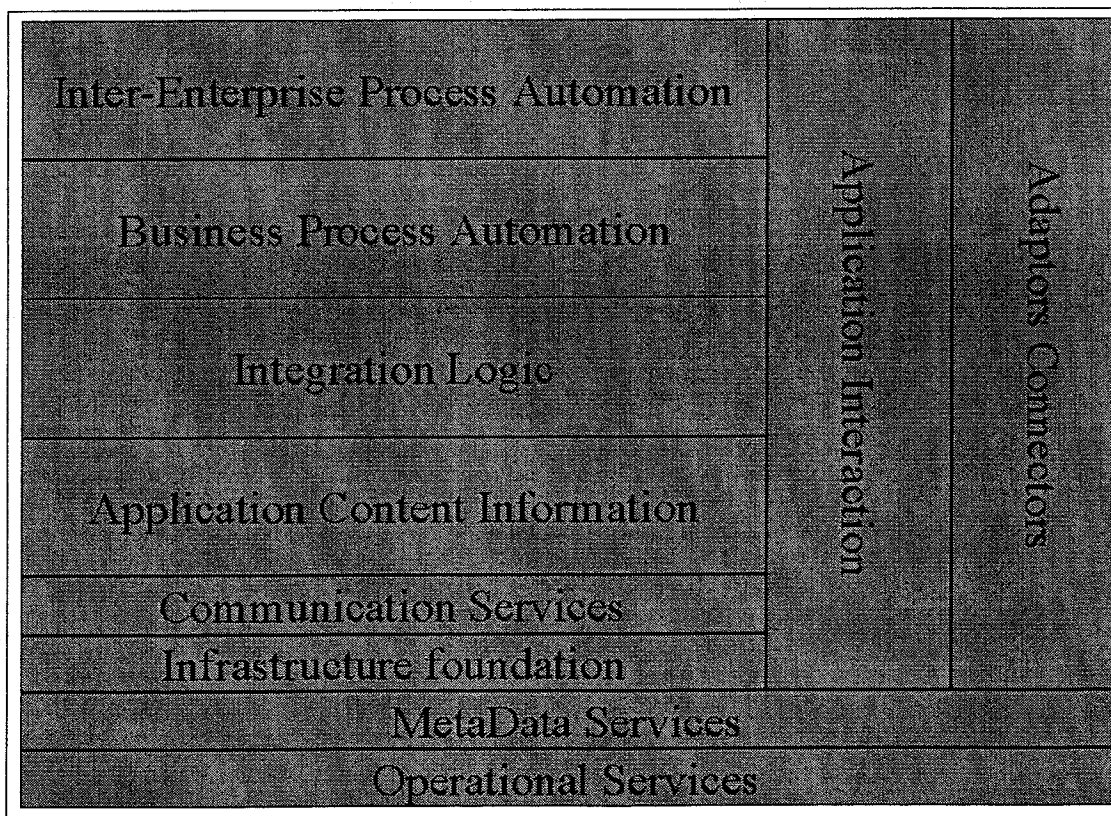
De laatste dienst werd opgenomen om de overige taken die met het wel en wee van een systeem te maken hebben onder te brengen. Men moet hierbij denken aan eigenschappen die voor een EAI belangrijk zijn zoals veiligheid en performance. Veel aanbieders hebben bijzondere oplossingen en toepassingen die bijvoorbeeld het systeembeheer of het fourterstel ondersteunen om deze verschillen in kaart te brengen werd de dienst *Operational services* toegevoegd.

9.4.2 De componenten

De aanbieders van EAI bieden producten die een of meerdere EAI diensten implementeren. Deze diensten bieden een basis om de verschillende aanbieders met elkaar te vergelijken.

In het onderstaande model worden de diensten zoals hierboven afgeleid, die een EAI kunnen bieden afgebeeld. Een bedrijf zal bij het integreren vaak de nadruk leggen op één of meerdere diensten. Op basis hiervan kan een bedrijf een keuze maken voor een bepaalde EAI oplossing. Zoals uit de motivatie voor de bepaling van de diensten bleek zit er een hiërarchie in de diensten. In afbeelding X is het model te zien. De diensten die de directe interactie met andere applicaties verzorgen (*Application Interaction* en *Adaptors/Connectors*) zijn aan de rechter kant geplaatst zodanig dat ze direct op elk niveau contact kunnen hebben. Sommige applicaties worden altijd op data niveau geïntegreerd anderen weer op proces niveau. De diensten *Metadata Services* en *Operational Services* zijn aan de onder kant geplaatst om aan te geven dat zij door alle andere diensten het zij direct het zij indirect gebruikt worden.

Hierna zullen de diensten kort toegelicht worden en zullen enkele criteria die men aan een dienst kan stellen geformuleerd worden.



Afbeelding 16

Inter-Enterprise Automation

Deze dienst wordt ook wel de *Business to Business* (B2B) dienst genoemd. Deze dienst maakt het mogelijk systemen van verschillende bedrijven met elkaar te laten communiceren. Een typisch scenario is de mogelijkheid automatisch een order te plaatsen als de voorraden lager dan een bepaald niveau zijn. In sommige gevallen weet het systeem al welke aanbieder van het product te benaderen in andere gevallen wordt een aanvraag op een zogenaamde *marketplace* gedaan. Hier worden de voorwaarden van verschillende aanbieders vergeleken.

Mogelijke criteria zijn:

- *partner management, transactie analyse en registratie*
- *het lokaliseren van aanbieders.*

Business Process Automation

Deze dienst definieert hoe de bedrijfsprocessen van een systeem geïmplementeerd zijn. Dit behelst de mogelijkheid om bedrijfsprocessen te modelleren, normaliter met behulp van een grafisch pakket. Maar ook het uitvoeren van het bedrijfsproces en het terugkoppelen van realtime bedrijfsinformatie, zoals bijvoorbeeld het aantal al verwerkte orders.

De meeste EAI aanbieders vullen deze functionaliteit in met een workflow-managementtool waar bedrijfsprocessen met behulp van automata of petrinetten beschreven wordt.

Als een applicatie een bericht verstuurt komt het systeem in een andere toestand, in deze toestand kunnen weer berichten ontvangen of verstuurd worden waardoor het systeem weer in een nieuwe toestand komt. Een EAI tool moet de mogelijkheid bieden om vele individuele sessies simultaan te verwerken. De verwerking van elke individuele order moet gemonitord worden inclusief alle berichten die bij die orders behoren.

Mogelijke criteria kunnen zijn:

- *gebruiksgemak van de modelleertools;*
- *de mogelijkheid om te monitoren;*
- *het debuggen van de modellen terwijl ze uitgevoerd worden;*
- *de mogelijkheid meerdere sessies te gelijktijd te laten verlopen;*
- *ondersteuning voor fout analyse en de analyse van het model, met informatie over hoeveel sessies verwerkt worden en de mogelijkheid sessiegevijs af te breken gebaseerd op specifieke criteria .*

Integration Logic

Deze dienst biedt enkele van de lagere functies die nodig zijn voor de erboven liggende Business Process Automation laag. De bedrijfsmodellen moeten berichten kunnen ontvangen van applicaties, de content kunnen begrijpen en deze content van de berichten gebruiken om beslissingen te nemen.

Mogelijke criteria kunnen zijn:

- *De mogelijkheid om de content van de berichten te bekijken;*
- *Gebruiksgemak;*
- *Performance*

Applicatiecontent transformatie

Elke applicatie die verbonden is met de EAI component heeft zoals besproken zijn eigen data model. Het gehele systeem presenteert weer een andere kijk op data. De vertaling van het ene naar het andere data model is een van de sleutelfunctionaliteiten van EAI

Deze dienst heeft meerdere taken. Elke applicatie zal unieke sleutels, meestal ID-nummers, aan objecten toekennen in de database van de applicatie. Zo kan een CRM systeem de id '3242df' aan een klant toekennen terwijl het factureringspakket '823782' aan deze zelfde klant toekent. Het EAI pakket moet de mogelijkheid bieden deze id's te vertalen zodanig dat er gecommuniceerd kan worden.

Berichten moeten samengevoegd worden of in stukjes opgedeeld worden. Een bericht kan meerdere berichten bevatten, de EAI moet dit bericht opdelen in uitvoerbare berichten die naar de desbetreffende applicaties gestuurd dienen te worden.

Mogelijke criteria kunnen zijn:

- *beschikbaarheid grafische tool om vertalingen te definiëren;*

- *ondersteuning voor bekende formaten;*
- *flexibiliteit om de tool uit te breiden om exceptionele gevallen te verwerken;*
- *data validatie en id mapping*

Communicatie diensten

Deze diensten beschrijven de lagere communicatie diensten die beschikbaar zijn voor de applicaties. Zoals vermeldt gebruiken de meeste E.A.I oplossingen een asynchroon berichtensysteem voor de communicatie. Een transactie wordt dan gedefinieerd door een reeks van berichten. Een EAI systeem moet waarborgen bieden dat alle berichten in een transactie in de juiste volgorde geleverd worden en dat de transacties alleen worden uitgevoerd indien alle berichten volledig zijn ontvangen.

Mogelijke criteria kunnen zijn:

- *performance van berichtensysteem;*
- *schaalbaarheid;*
- *transactie-ondersteuning.*

Infrastructural foundation

Deze dienst beschrijft de eisen aan de operationele omgeving van een EAI Dit bevat de hardware, de besturingssystemen, de netwerk protocollen enzovoorts.

Mogelijke criteria kunnen zijn:

- *de pakketten die worden ondersteund;*
- *ondersteuning voor de verschillende netwerkprotocollen.*

Application interaction

Deze dienst regelt hoe applicaties met elkaar communiceren. Zoals bekend zijn er meerde mogelijkheden voor applicaties om met elkaar te communiceren; een keuze wordt gemaakt naar aanleiding van de eisen aan de communicatie. De vormen worden ingedeeld in 'publish/subscribe', 'request/respons' en 'data push'.

Mogelijke criterium kan zijn:

- *het aantal verschillende communicatie modellen die er geboden worden*

Application Adaptor/Connectors

Deze dienst definieert hoe applicaties met het EAI systeem verbonden dienen te worden. Hoe deze functioneren en welke verschillen er zijn werd beschreven in hoofdstuk 8.3.

Mogelijke criteria zijn:

- *Beschikbaarheid van de verschillende connectoren;*
- *Performance van connectoren;*
- *De omgeving waarin ze ontwikkeld werden*

Metadata diensten

Een EAI project vergt veel organisatie. Vaak is er een bedrijfsbreed datamodel dat door alle applicaties gebruikt dient te worden. Het datamodel moet gedefinieerd en opgeslagen worden in het EAI pakket. Alle proces modellen, vertaaltabellen en overige data dienen ook opgeslagen te worden. De meeste EAI oplossingen bieden de mogelijkheid deze metadata op te slaan en makkelijk benaderbaar te maken via een user interface. Een pakket moet geïntegreerd kunnen worden met bestaande metadata oplossingen zoals LDAP en BizTalk. LDAP is een verzameling protocollen voor de toegang tot adres informatie, gebaseerd op een eenvoudige versie van de X.500 standaarden. Integrestelling tot X.500 ondersteunt LDAP TCP/IP, wat nodig is voor de internet toegang

Mogelijke criteria zijn:

- *Mogelijkheid om metadata op te slaan;*

- *Mogelijkheid om meerdere modellen op te slaan;*
- *vertalingen, connectoren, integratie met LDAP en integratie met XML schema's.*

Operationele diensten

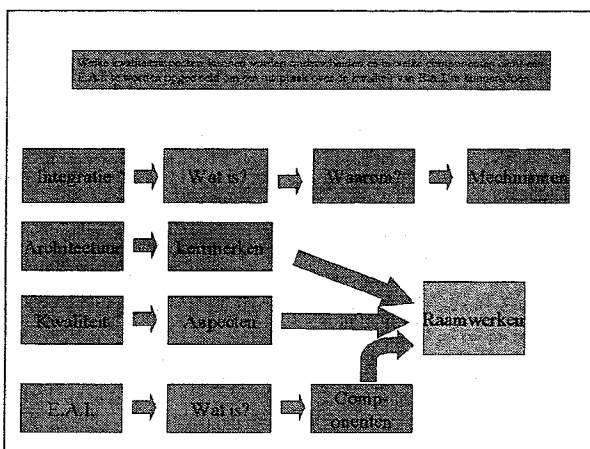
Een EAI wordt het centrale punt van een veelal complex systeem. Daarom moet een EAI makkelijk te beheren zijn door de desbetreffende systeembeheerder. Het systeem moet veiligheid bieden zodat de wachtwoorden die applicaties gebruiken niet toegankelijk zijn voor ongeautoriseerd personeel. Maar ook ander veiligheidsdiensten moeten hier geregeld worden; de verschillende aspecten van veiligheid worden in detail in Appendix D beschreven.

Performance en betrouwbaarheid zijn twee andere kritische aangelegenheden voor een EAI. Het systeem moet schaalbaar zijn en een vorm van *load balancing* moet ondersteund worden. Het systeem moet kunnen *hotswappen*; het omschakelen naar een ander systeem in het geval van een systeem crash. Integratie met de al beschikbare monitoring en netwerk beheer pakketten is gewenst. Backups van de applicatiedata moeten worden verzorgd en het moet eenvoudig zijn fouten in het systeem op te sporen.

Mogelijke criteria zijn:

- *gebruiksgemak van de administratieve en beheers voorzieningen;*
- *de mogelijkheid om te load balancen;*
- *mogelijkheid tot fouterstel;*
- *integratie met populaire netwerk monitoring tools en ondersteuning voor live backups.*

9.4.3 Kwaliteitsaspecten in relatie tot de componenten



Leeswijzer

Nadat op basis van de diensten een model is neergezet zullen uitspraken gedaan moeten worden over de componenten in relatie tot de kwaliteitsaspecten. Zoals in hoofdstuk 6 uiteengezet werd zijn deze aspecten onder te verdelen in zeven groepen

In Appendix B worden deze begrippen gedefinieerd en in deelaspecten opgedeeld om ze handelbaarder te maken.

In eerste instantie zou verondersteld kunnen worden, aangezien de componenten op zich zelf applicaties kunnen zijn, ze aan alle kwaliteitsaspecten moeten voldoen. Dit is correct maar vanuit het standpunt inneemt dat men de kwaliteit van de gehele oplossing wilt beoordelen, draagt een bepaalde component aan een bepaald kwaliteitsaspect meer bij dan aan een ander kwaliteitsaspect.. Met andere woorden men voor elke component in het kader van de gehele oplossing de meer of minder relevante kwaliteitscriteria vaststellen.

Zo zal een goed onderliggend netwerk de efficiency ten goede komen. De efficiency wordt ook verbeterd als processen makkelijk gemodelleerd kunnen worden. Om hier een keuze in te kunnen maken is voor volgende strategie gekozen: voor elke component bepalen we arbitrair de drie kwaliteitsaspecten van de totaaloplossing die het meest beïnvloedt worden door deze component. In het voorbeeld kan dan ook gezegd worden dat een verandering in het onderliggende netwerk minder efficiency voor- of nadelen biedt dan het optimaliseren van het proces modelleren, omdat daar de productiviteit, zuinigheid en herbruikbaarheid sterker door veranderen. Een dergelijke afweging zal nu voor alle componenten worden gemaakt.

Inter-Enterprise Proces Automation;

Gekozen kwaliteitsaspecten: Effectiviteit en Exclusiviteit

Motivatie: Een van de doelstellingen van applicatie integratie is het beter samenwerken met leveranciers en klanten. Door het systeem open te stellen wordt dus de mate waarin de oplossing overeenstemt met de eisen en doelstellingen van de gebruiker verhoogt. Daarnaast speelt ook de exclusiviteit een belangrijke rol, men laat immers 'vreemden' toe op de eigen applicaties. Deze vreemden dienen goed gevolgd te kunnen worden en eventueel naderhand, bijvoorbeeld in geval van fraude, achterhaald te kunnen worden daarom wordt als derde object voor controleerbaarheid gekozen.

Business Process automation

Gekozen kwaliteitsaspecten: Efficiency, Continuïteit en Integriteit.

Motivatie: Efficiency werd gekozen omdat duidelijk is dat een goede modellering van de processen de afhandeling door de applicaties ten goede komt, de productiviteit van het hele systeem gaat er op vooruit. Deze component kan de mogelijkheid bieden processen concurrent af te handelen wat ook de efficiency voordelen brengt. Daarnaast is gekozen voor de continuïteit omdat hier de fout- en modelanalyse plaats vindt. Dit moet kunnen zonder lopende processen te onderbreken. De optie voor deze analyse is ook de motivatie voor het derde aspect waarvoor gekozen is: de integriteit. Hoe beter de modelleertechniek en de grafische representatie hoe hoger de mate waarin het systeem in overeenstemming is met de afgebeelde werkelijkheid.

Integration logic

Gekozen kwaliteitsaspecten: Beheersbaarheid, Efficiency, Effectiviteit

Motivatie: Hier wordt bepaald hoe makkelijk koppelingen gemaakt kunnen worden en hoe de performance zal uitpakken. Performance is een deelaspect van de beheersbaarheid. Door deze dienst worden de berichten voorbereid en is dus ook een efficiency 'bottleneck', gebeurt dit namelijk niet slim dan gaan er veel instructies verloren. Daarnaast is gekozen voor effectiviteit omdat de eis aan zo een systeem toch zeker is om de verschillende applicatie goed aan elkaar te verbinden. Werkt de *Integration Logic* niet zoals het hoort dan doet het systeem niet waarvoor het aangeschaft werd, namelijk het integreren.

Application content transformation

Gekozen kwaliteitsaspecten: Integriteit, Continuïteit en Controleerbaarheid.

Motivatie: Bij de transformatie is het van belang dat zeker wordt gesteld dat deze succesvol plaatsvinden; daarom hoort daar het kwaliteitsaspect integriteit bij. Men wil immers waarborgen hebben en deze later ook kunnen controleren. Zodoende wordt de controleerbaarheid als tweede aspect toegevoegd. Met het oog op de toekomst moeten nieuwe dataformaten eenvoudig toegevoegd kunnen en wordt de keuze gemaakt voor continuïteit.

communication services en Infrastructural foundation

Gekozen kwaliteitsaspecten: Efficiency, Continuïteit en Integriteit.

Motivatie: Een hogere performance leidt tot een hogere productiviteit en dat kan weer leiden tot een hogere efficiency. Maar een hogere performance kan leiden tot risico's met betrekking tot de waarborging en nauwkeurigheid van de gegevens.

Daarom wordt de integriteit ook beïnvloed door het onderliggende berichtensysteem. Daarnaast is de schaalbaarheid cruciaal voor een berichten systeem. Schaalbaarheid is een maatstaf voor hoe goed een systeem zijn taken kan blijven vervullen als er meer te verwerken valt. Dit valt dus onder het kopje continuïteit. Met betrekking tot de onderliggende netwerk structuur (in het model de *Infrastructural Foundation*) kan het zelfde betoog gehouden worden en dat resulteert in de zelfde drie meest relevante kwaliteitsaspecten.

Application interaction

Gekozen kwaliteitsaspecten: Efficiency, Continuïteit en Integriteit.

Motivatie: Het is duidelijk dat de manier van communiceren betrekking heeft op de efficiency van het systeem. Door manier van afleveren van berichten en de eventuele bevestiging of niet wordt ook de integriteit beïnvloed. Als laatste wordt gekozen voor de continuïteit, het is namelijk zo dat er communicatie vormen zijn waarbij op een eenvoudige wijze elementen toegevoegd of veranderd kunnen worden dan bij andere modellen; zeker met betrekking tot de eenvoud en het effect dat het op het hele systeem heeft zijn er grote verschillen.

Adaptors Connectors

Gekozen kwaliteitsaspecten: Effectiviteit, Continuïteit en Beheersbaarheid.

Motivatie: Door een breed scala aan connectoren te bieden wordt de bruikbaarheid van het systeem verhoogd. Een breed scala biedt immers meer mogelijkheden om verschillende applicaties met elkaar te laten interacteren. Hierdoor sluit het systeem beter aan bij de doelstelling: het integreren van applicaties. Het eerste aspect is dan ook effectiviteit. Daarnaast kan het eenvoudig of moeilijk kunnen maken van nieuwe adaptoren in de toekomst belangrijk zijn. Wil men nieuwe applicatie ook snel kunnen opnemen moet dit goed geregeld zijn, vandaar dat ook continuïteit ook als relevant wordt aangemerkt. Als laatste wordt gekozen voor de beheersbaarheid. Een deelaspect van beheersbaarheid is de connectiviteit. Hoe meer verschillende adaptoren er zijn hoe hoger de connectiviteit.

Metadata Services

Gekozen kwaliteitsaspecten: Integriteit, Controleerbaarheid en Beheersbaarheid.

Motivatie: Een belangrijke eis aan (meta)data is dat er van uitgegaan kan worden dat de data juist is, oftewel dat ze in overeenstemming zijn met hetgeen wat men wilde beschrijven. Daarom nemen we integriteit als relevant kwaliteitsaspect. Daarnaast biedt het hebben van zo'n data model veel mogelijkheden op het gebied van de beheersbaarheid. Zo wordt de onderhoudbaarheid verhoogd en kan de connectiviteit veranderen naar mate de vertaalslag van een data type naar het datamodel eenvoudig of lastig is. Ook is het belangrijk dat de interne structuur er zich voor leent bepaalde kengetallen te meten en tests uit te voeren. Het derde aspect is dan ook controleerbaarheid.

operationele diensten

Gekozen kwaliteitsaspecten: Beheersbaarheid, Continuïteit en Exclusiviteit.

Motivatie: Operationele diensten kenmerken zich door hun ondersteunend karakter.

Een aantal onderdelen hiervan zijn de beveiliging, integratie met andere controletools en systeembeheerpakketten; daarom wordt als een relevant kwaliteitsaspect de beheersbaarheid genoemd. Daarnaast zijn veel EAI oplossingen in de operationele diensten uitgebreid met mechanismen voor fouttolerantie, back-up systemen en *load balancing* dit betekent dat systeem niet snel vast loopt en als dit gebeurt het weer snel herstelt kan worden; dit heeft betrekking op de continuïteit. De beveiligingsmechanismen, de login controle en de administratieve tools worden gebruikt om in de gaten te houden wie wat doet en of dit door geautoriseerde persoon is gebeurd; daarom wordt als derde kwaliteitsaspect voor de exclusiviteit gekozen.

9.4.4 Het model in de praktijk.

Het gereedschap dat werd ontwikkeld heeft slechts waarde als het voor de opdrachtgever in de praktijk toepasbaar is. Daarvoor moeten er enkele aspecten van het model en het gebruik ervan worden toegelicht. Hieronder volgen enkele toepassingsgebieden en voorts wordt beschreven in welke situaties ondernemingen, of andere belanghebbenden, die geïnteresseerd zijn in een matrix als deze en in welke vorm de resultaten gepresenteerd kunnen worden.

Toepassingsgebieden

Analyse

Door een software-oplossing te verdelen in componenten en deze te beschrijven en te beoordelen wordt het systeem en zijn functies duidelijk vastgelegd. Hierdoor is het mogelijk, via het maken van een afbeelding van een oplossing op het model, toe te lichten waar bepaalde zaken geregeld worden. Hiermee wordt bedoeld dat na een analyse van een systeem, beschreven wordt welke componenten van de software op elk van de componenten van het model afgebeeld worden.

Stel dat men weten wil welk datamodel er gehanteerd wordt, kan men door na te gaan welke component van het onderzochte product hiervoor verantwoordelijk is, snel een antwoord vinden. Tevens krijgt men inzicht in de architectuur van een product en uit welke deelproducten het is opgebouwd.

Het model is dus geschikt voor een snelle analyse van een oplossing.

Vendor selection

Entiteiten die een EAI oplossingen willen invoeren zoeken de geschikte leverancier voor deze software: het proces van het vergelijken van oplossingen met elkaar wordt 'vendor selection' genoemd. De kwaliteit van een product speelt daarbij uiteraard een belangrijke rol. Er zijn echter ook andere aspecten zoals: de aanschafkosten, de invoerkosten, de tijd die het kost om zo een systeem in te voeren, de servicegraad van de aanbieder en ook met emotionele factoren wordt rekening gehouden: zoals bijvoorbeeld: sluit de filosofie van de bedrijven aan, is er een goed gevoel over de continuïteit van de leverancier, enzovoorts.

Een lijst van aandachtspunten bij het selecteren van een aanbieder is die van Mike Gilpin [REF 13]. Deze lijst bevat naast de kosten van gebruik en invoering aandachtspunten zoals: type modelleergereedschappen, flexibiliteit, ondersteuning voor verschillende omgevingen, type 'messaging', uitbreidbaarheid, aantal connectoren en integratie van legacy. Het model gaat in op deze punten en biedt dus een goede start voor het kiezen van een leverancier.

Kwaliteitsoordeel

Het model is maakt een tussen een kwaliteitsmodel en het dienstenmodel. Het is een raamwerk ter beoordeling van een EAI omgeving. De vraag is nu hoe men een oordeel kan vellen over een gegeven oplossing en in welke vorm dit gebeurt.

Uitgangspunt wordt: voor elke component geeft men een cijfer van een tot en met tien. Deze manier van werken kan leiden tot een subjectief oordeel. In het onderzoek wordt dan vaak met oplossingen gewerkt waarbij meerdere mensen zo een matrix uitwerken en men een gewogen gemiddelde als resultaat neemt. Het uitkiezen van een goed panel is dan essentieel. Men kan volgende methoden onderscheiden:

Domeingebaseerde aanpak

Zoals in hoofdstuk 6 beschreven werd kan men een informatiesysteem vanuit verschillende domeinen beoordelen. Er is daar uiteengezet dat de domeinen 'informatiesysteem', 'technisch systeem' en 'processysteem' het meest van toepassing zijn.. Een mogelijkheid is dan ook: de matrix vanuit deze domeinen te beoordelen en dan een gemiddelde te bepalen. Daarnaast kan men ook de aard van het proces kunnen differentieren zoals tussen het beheer en het gebruik.

Op de gebruiker gebaseerde aanpak

Een methode waarvan vaak gebruik gemaakt wordt bij het beoordelen van gegevens is er meerdere personen met verschillende achtergronden naar te laten kijken. Aangezien een EAI verschillende bestaande oplossingen aan elkaar knoopt zou men de experts van de deelsystemen naar heel het systeem kunnen laten kijken en daarna via een gewogen gemiddelde tot een eindconclusie te komen.

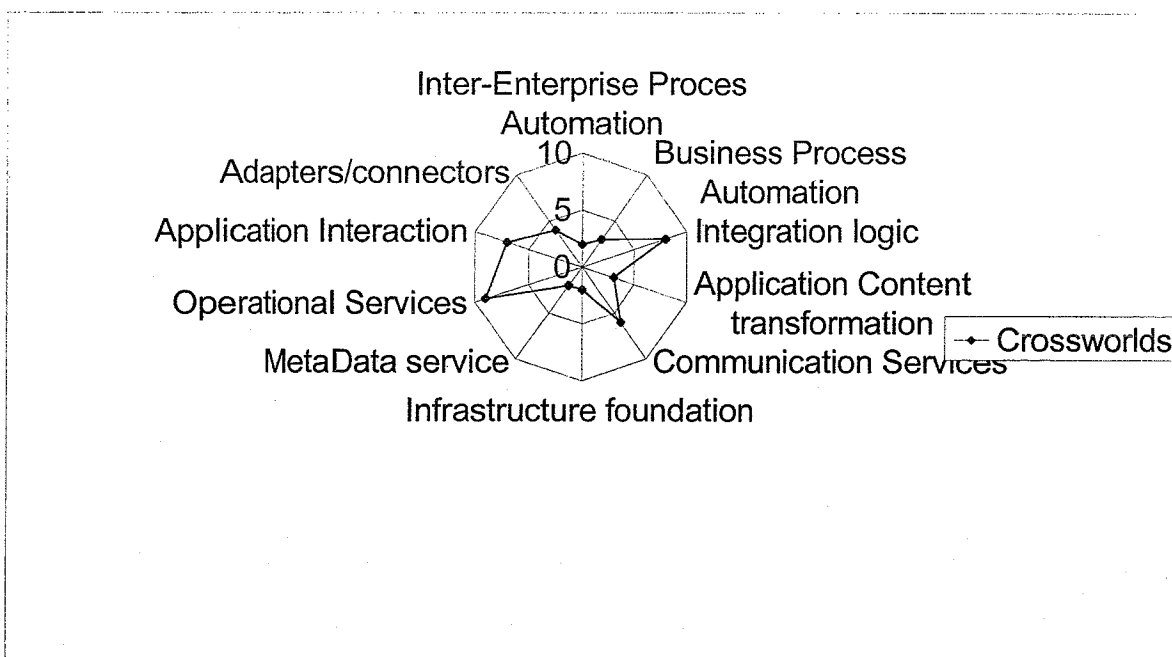
Op het type onderneming gebaseerde aanpak

Verder is het ook zo dat verschillende typen ondernemingen verschillende eisen aan software stellen. Zo zal een handelsonderneming bepaalde objecten meer belang toe dichtten dan een onderneming uit de procesindustrie. Dit komt omdat ze verschillende doelstellingen hebben en de kwaliteit van een software-product ook wel de mate is waarin een product voldoet aan de eraan gestelde eisen, welke voortvloeien uit het gebruiksdoel. Dit wetende zou men in de matrix de objecten verschillende gewichten kunnen geven en zo zou het oordeel van een gegeven oplossing per type onderneming kunnen verschillen. Dit houdt in dat men elk resultaat metn een getal vermenigvuldigd en wel zo dat de som van het aantal factoren weer het aantal componenten is.

Rapportage

Indien er gekozen is voor een bepaalde aanpak is het van belang dat de resultaten zo overzichtelijk mogelijk gepresenteerd kunnen worden. Omdat de resultaten snel toegankelijk en door een advies-bureau bruikbaar moeten zijn; is er voor de volgende rapportage gekozen:

- ten eerste wordt een systeem en zijn componenten afgebeeld op de in paragraaf 9.4.beschreven componenten model. De toelichting op de componenten wordt in het begeleidende stuk tekst van de rapportage gegeven en is gebaseerd op de door de bedrijven beschikbaar gestelde informatie en de door de audit praktijk zelf verzamelde informatie en ervaringscijfers;
- daarna worden de beoordelingen van de componenten in een radar grafiek gepresenteerd zoals in afbeelding 17 te zien is. Dit wordt gedaan zo dat men snel inzicht kan krijgen in wat de relatief sterke en de zwakkere punten zijn.



Afbeelding 17

In de rapportage moeten de meetbare technische kenmerken en eventuele beschikbare procesbeschrijvingen worden opgenomen. Hierbij moet worden gedacht aan een opsomming van de adaptoren en waarvoor ze dienen, en een case study met procesbeschrijving. De uiteindelijke rapportage zal een duidelijke, beknopte doch bruikbare indruk van het systeem moeten achterlaten. In appendix H wordt een voorbeeld van een uitgewerkt rapport gegeven.

9.4.5 Conclusie en vervolgonderzoek

Het model kan dienen als referentiekader. De basis van het referentiekader is het inzichtelijk en toegankelijk maken van een geïntegreerde omgeving met een EAI door het te beschrijven in een logisch model. Niet alleen kan aan de hand van dit model de werking worden duidelijk gemaakt maar tevens biedt dit model een manier, om op een gestructureerde wijze een beoordeling uit te voeren. In deze studie is er voor gekozen de beoordeling te richten op onderwerp kwaliteit en wel specifiek de kwaliteit zoals deze in de audit praktijk van PwC wordt gehanteerd. Ondanks dat de kwaliteitsaspecten in deze studie niet universeel zijn, kunnen deze wel als 'de facto' standaard binnen de auditpraktijk worden gezien. De kwaliteitsaspecten vormen daarmee een geschikt instrumentarium voor de beoordeling van de EAI door auditfirma's. Dit wil niet zeggen dat dit referentiekader de absolute eenduidige meetlat vormt. Inherent aan het begrip kwaliteit is dat de feitelijke invulling altijd situationeel wordt bepaald. Dit neemt niet weg dat het referentiekader een prima startpunt is voor de beoordeling van kwaliteit van EAI. Op basis van de specifieke situatie en de mate van relevantie kan het referentiekader worden toegesneden op de feitelijke behoefte en worden ingezet voor de kwaliteitsbeoordeling van EAI. Als mogelijke praktijktoepassingen is te denken aan: Vendor Selection, pakket vergelijking, system-audits, pre- en postimplementatieaudit en analyse. Dit soort praktijktoepassingen in de toekomst waarschijnlijk belangrijker worden. Redenen zijn ondermeer: de toenemende vraag naar integratie van systemen, dit niet alleen vanuit de groeiende business complexiteit maar ook op grond van economische motieven. Bijvoorbeeld het hergebruik van (legacy) systemen. De toenemende aandacht voor kwaliteit en met name de borging daarvan, kan men toeschrijven aan de centrale plaats die het inneemt maar vormt daarmee ook een 'Single point of Failure'. Gelet op het cruciale karakter zal het management de met EAI samenhangende risico's terug willen brengen binnen de grenzen van hun eigen risicoprofiel.

Een onafhankelijke en onpartijdige beoordeling wordt daarbij gezien als een welkome ondersteuning. Toch dient opgemerkt te worden dat er ruimte is voor vervolgonderzoek. Zo zou men de eisen vanuit een bepaalde industrie veel nauwgezetter in kaart kunnen brengen. De risico's van een oplossing met EAI kunnen dan zowel functioneel alsook commercieel beter in kaart gebracht worden. Dit zou men kunnen doen door een model af te leiden dat beter kwantificeerbaar is.

In deze studie wordt vooral van de hoogniveau aanpak gebruik gemaakt. De werking van het systeem op object niveau werd niet beschreven, ook dit zo een vervolgonderzoek waard zijn. Verder lijkt het mogelijk meer functionele eisen te stellen waardoor men een zeer uitgebreide checklist zou kunnen samenstellen.

Een interessante studie is ook het vergelijken van de verschillende gebruikte onderliggende software en standaarden. Bijvoorbeeld een vergelijking tussen EAI met CORBA en EAI met COM+.

Ten slotte valt nog op te merken dat men ook het kwaliteitsmodel fijnmaziger kan definiëren en daardoor beter kwantificeerbare oplossingen zou kunnen maken. Hierbij zou men kunnen denken aan benchmarks voor bijvoorbeeld de performance.

Referenties

- [1] Gartner. Gartner Group: Report Forcasts for E-markets 2000
- [2] Silver, Pyke en Peterson: Inventory Management and Production Planning and scheduling, third edition
- [3] NOREA: Nederlandse orde van Register EDP auditors, Geschrift 1, It auditing aangeduid
- [4] J.M.Juran: Juran's Quality Control Handbook
- [5] Phil Crosby: Getting it right
- [6] Nederlands Normerings Instituut: ISO Normering
- [7] Garvin: Managing quality: the strategic and competitive edge.
- [8] Brown: Assessing the quality of large, Software intensive Systems, a case study
- [9] Boehm: Megaprogramming
- [10] Gartner, Gartner Group: Data in its place
- [11] Zahavi: Architectural design principles
- [12] Zahavi: Enterprise Application Integration with CORBA
- [13] Fowler: Analysis Patterns: Reusable Object Models
- [14] Giplin: How to select an EAI solution

Appendix A Domeinen en objecten

Informatie strategie

De informatie strategie wordt omschreven als het geheel van doelstellingen, uitgangspunten en randvoorwaarden voor het omgaan met informatie binnen een organisatie en voor de organisatie van de informatievoorziening zelf.

Objecten bij het domein informatiestrategie:

- Informatiebeleidvorming
- Informatieplanning
- Informatie architectuur
- Interne en externe afstemming
- Research

Informatiemanagement en informatietechnologie management (IM/IT-management)

IM/IT-management omvat de door de leiding van een organisatie te scheppen voorwaarden voor de ontwikkeling, het beheer en het gebruik van geautomatiseerde systemen, alsmede de besturing van deze processen, zodanig dat de leiding kan vaststellen dat een de door haar in de informatiestrategie geformuleerde doelstellingen en randvoorwaarden voldaan wordt.

Objecten bij het domein IM/IT-management

- Plannen
- Organiseren
- Budgetteren
- Bemannen
- Beslissen
- Coördineren
- Rapporteren
- Innoveren
- Communiceren
- Controleren
- Besturen
- Motiveren
- Kennis vergaren

Informatiesystemen

Onder een informatiesysteem wordt verstaan de geautomatiseerde processen die primair ontworpen zijn om de mens te voorzien van de gegevens, dan wel om de mens in staat te stellen de gegevens opgeslagen in computers- en overdraagbaar via datacommunicatietechnieken- te creëren, muteren, verwijderen, verspreiden en/of anderszins te manipuleren. Het geheel van de organisatie en hulpmiddelen die primair dienen voor de ontwikkeling en het gebruik van informatiesystemen behoort tot het domein van informatie systemen.

Objecten bij het domein informatiesystemen

- Ontwikkelorganisatie
- Onderhoudsorganisatie
- Projectorganisatie
- Kwaliteitscontrole
- Planning en control
- Overige maatregelen projectbeheersing
- De implementatieplanning en de beheersing hiervan
- Technisch beheer
- Versiebeheer

- Distributie management
- Hulpmiddelen
- Eisen aan het informatie systeem
- Applicatie-architectuur
- Databases
- Operationeel gebruik
- Functioneel beheer
- Opdrachtgever/eigenaar
- Het informatiesysteem

Technische systemen

Systemen die ontworpen zijn om geïmplementeerd te worden in hardware en systeemprogrammatuur met het doel de hardware en/of (andere systeem-) programmatuur aan te sturen.

Technische systemen ondersteunen de geautomatiseerde processen binnen de informatiesystemen en processystemen door het aansturen en geautomatiseerd beheersen van de hardware die deel uitmaakt van de technische infrastructuur.

Objecten die behoren bij technische systemen

- Voorbereiding/ontwikkelorganisatie
- Onderhoudsorganisatie
- De kwaliteitscontrole op het technisch beheer
- Planning en control
- Configuratie beheer
- Versiebeheer
- Capaciteitsbeheer
- Technisch beheer
- Beheer van de technische infrastructuur
- Distributiemanagement
- Tweede lijns ondersteuning rekencentra
- De hulpmiddelen en technieken voor het gebruik door de technische beheerders
- Eisen aan het technisch systeem
- Functioneel beheer
- Het technisch systeem

Processystemen

Processystemen zijn systemen die ontworpen zijn om elektronische interfaces en daarmee apparaten aan te sturen.

Objecten van het domein processystemen

- De interne organisatie van de professionele ontwikkelorganisatie
- Organisatie voor het onderhoud aan professioneel beheerde systemen
- Projectorganisatie
- Kwaliteitscontrole
- Planning en controle van professionele ontwikkelingen
- De implementatieplanning en de beheersing hiervan
- Technisch beheer
- Versiebeheer
- Configuratie management
- Beheer technische infrastructuur
- Het operationele beheer van processystemen
- Hulpmiddelen
- Eisen aan het processysteem
- Overige objecten betreffende processystemen in ontwikkeling

- Applicatie-architectuurfunctie
- Database administratiefunctie
- Opdrachtgever/eigenaar
- De organisatie van het functioneel beheer van de processystemen
- Het processysteem

Operationele ondersteuning

Onder operationele ondersteuning wordt verstaan: alle activiteiten van een organisatie die gericht zijn op het beheren en beschikbaar houden van de technische infrastructuur en de onder beheer zijnde IT-processen, conform de afgesproken standaarden en service level agreements, alsmede de administratie daarvan.

Objecten van het domein operationele ondersteuning

- De organisatie van het operationele beheer
- De interne organisatie voor het beheersen van de operationele ondersteuning
- Interne controle
- Beveiliging
- Autorisatie
- Risico management
- De administratie van het rekencentrum
- Functioneel beheer technische systemen
- De productieplanning en de productie controle
- Nabewerking
- Operations van apparatuur
- Resource management
- Library management
- Netwerk operations
- Helpdesk
- Customer liaison
- Acceptatie van (veranderingen in) technische systemen
- De uitwijkregelingen (contingentie)
- Inkoop/verkoop hardware en daarbij behorende programmatuur
- Leveranciersondersteuning
- Capaciteitsprognoses en metingen van de faciliteiten
- Netwerk management
- Acces control beheer
- Change management
- Distributiemanagementfuncties
- De administratie van versies van systemen onder beheer
- Probleemmanagement
- Operationele ondersteuning van de bedrijfsafdelingen

Appendix B Kwaliteitsaspecten

Om misverstanden te voorkomen is het belangrijk om het begrip kwaliteit wat nader toe te lichten.

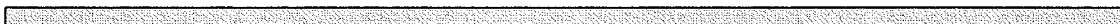
Naar het Norea geschrift nr. 1 over IT-auditing worden zeven kwaliteitsaspecten gedefinieerd :Effectiviteit, Efficiency, Exclusiviteit , Integriteit, Controleerbaarheid, Continuïteit en Beheersbaarheid. Om een inzicht te krijgen in wat er met deze begrippen bedoeld wordt verwijs ik naar afbeelding 1. Hierin ziet U de verschillende kwaliteitsaspecten met de eventuele deelaspecten die daarbij een rol spelen en hun definities:

Kwaliteits Aspect	Deel Aspect	Definitie
Effectiviteit		De mate waarin een object in overeenstemming is met de eisen en doelstellingen van de gebruikers en de mate waarin een object bijdraagt aan de organisatiedoelstellingen zoals die in de informatiestrategie zijn vastgelegd
	Bruikbaarheid	Is het systeem afgesteld op de cultuur van de organisatie
	Ondersteuning	Levert de informatie voorziening een bijdrage aan de vorming van het beleid en het nemen van besluiten op management niveau
Efficiency		De verhouding tussen de gerealiseerde en begrote kosten van een object
	Zuinigheid	het prestatieniveau in verhouding tot het verbruik van middelen uit de technische infrastructuur met de daarbij behorende kosten
	Herbruikbaarheid	de 'herbruikbaarheid' van delen van een informatiesysteem
	Productiviteit	'De mate waarin de productiviteit van gebruikers toeneemt'
Exclusiviteit		In hoeverre maken personen of apparatuur via geautoriseerde procedures en beperkte bevoegdheden gebruik van IT-processen?
	Autorisatie	De adequate delegatie van bevoegdheden

Kwaliteits Aspect	Deel Aspect	Definitie
	Authenticatie	de adequate verificatie van geïdentificeerde personen of apparatuur
	Identificatie	het mechanisme ter herkenning van personen of apparatuur
	Controle bevoegdheden	het geautomatiseerd vaststellen of geïdentificeerde personen of apparatuur de gewenste handelingen mogen uitvoeren



Integriteit		de mate waarin het object (gegevens en informatie-, technische- en processystemen) in overeenstemming is met de afgebeelde werkelijkheid
	Volledigheid	de mate van zekerheid dat het object volledig aanwezig is
	Nauwkeurigheid	overeenstemming van het aggregatieniveau van de presentatie met de werkelijkheid
	waarborging	de vraag of de correcte werking van de IT-processen is gewaarborgd



Controleerbaarheid		De mogelijkheid om kennis te verkrijgen over de structurering (documentatie) en werking van een object
	Testbaarheid	Hoe is de integere werking van programmatuur te meten?
	Meetbaarheid	Zijn er voldoende meet- en controlepunten aanwezig?
	Verifieerbaarheid	is de integere werking van een object te peilen?



Continuïteit		De mate van continue beschikbaarheid van een object en de ongestoorde voortgang van de gegevensverwerking
	beschikbaarheid	De mate waarin gegevens en IT-processen de organisatie op de momenten, die de organisatie heeft geëist, ondersteunen

Kwaliteits Aspect	Kwaliteits Aspect	Definitie
	Portabiliteit	De overdraagbaarheid van het informatiesysteem naar verschillende technische infrastructuren
	Herstelbaarheid	Het gemak en de snelheid waarmee de informatievoorziening hersteld kan worden na een storing
<hr/>		
Beheersbaarheid		De mate waarin het object kan worden aangestuurd en bijgestuurd, zodat het object bij voortdurend aan de daaraan gestelde eisen voldoet
	Onderhoudbaarheid	Het gemak waarmee correctieve, adaptieve en perfectieve wijzigingen zijn aan te brengen en waarmee de functionaliteit is aan te passen
	Connectiviteit	Het gemak waarmee koppelingen met andere systemen tot stand kunnen komen en het gemak waarmee onderdelen binnen een systeem op elkaar aansluiten en aangepast kunnen worden
	Effectiviteit	De efficiency en de tijdigheid van correctieve maatregelen - niet te verwarren met het eerder genoemde hoofd-kwaliteitsaspect effectiviteit
	Beveiliging	De bescherming van waarden tegen ongeautoriseerde en ongewenste ingrepen en calamiteiten alsmede de maatregelen om de gevolgen te minimaliseren

Deze zeven kwaliteitsaspecten kunnen op hun beurt ook weer uit verschillende domeinen benaderd worden. Meestal onderscheidt men volgende domeinen: 'domeinen'.

Maar wat is kwaliteit nu echt?

Er zijn, in de loop van de jaren, heel veel definities van het begrip 'kwaliteit' gegeven. Een greep daaruit:

- Fitness for use (geschiktheid voor gebruik) - Joseph Juran
- Conformance to requirements (voldoen aan specificaties) - Phil Crosby
- I know it when I see it
- Geheel van kenmerken van een entiteit dat betrekking heeft op het vermogen van die entiteit om kenbaar gemaakte en vanzelfsprekende behoeften te bevredigen - NEN EN ISO 8402

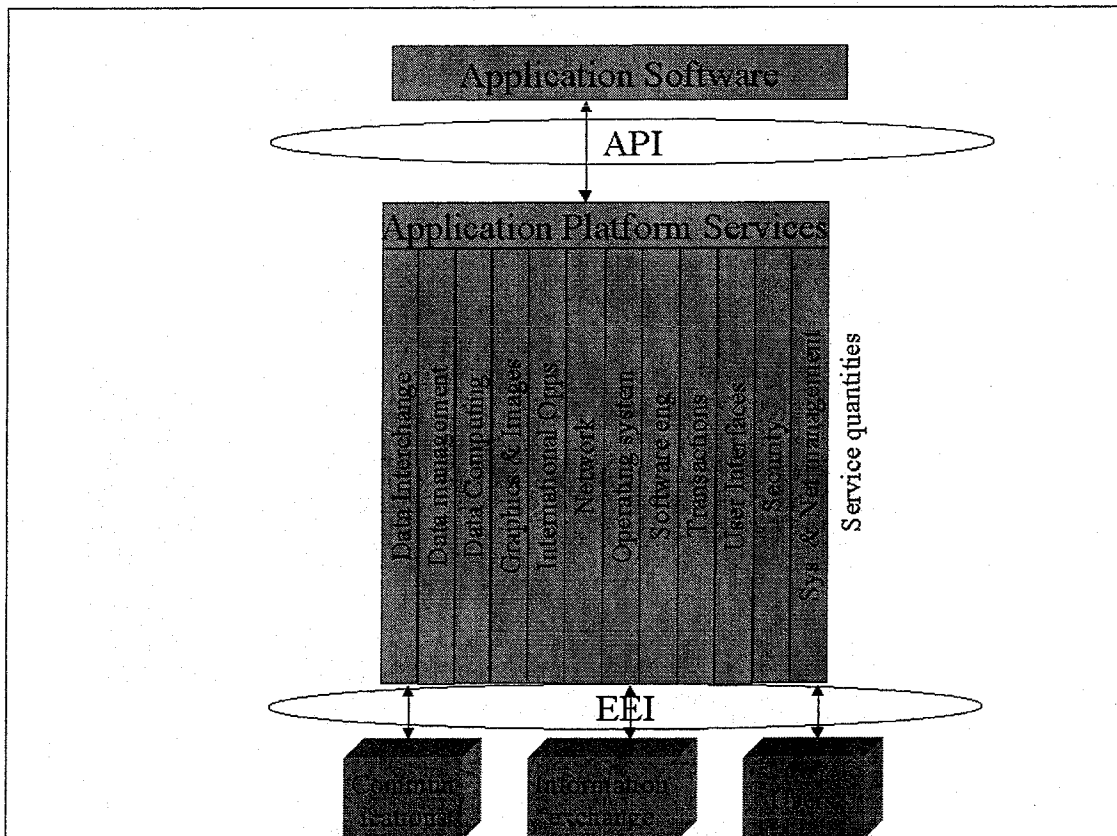
Al die definities bevatten een kern van waarheid, maar elk heeft ook z'n beperkingen in het gebruik. Een definitie van kwaliteit die in veel gevallen bruikbaar is gebleken (maar uiteindelijk natuurlijk dezelfde makkes heeft als de bovengenoemde) is deze:

Voldoen aan de verwachtingen van de klant

Appendix C Raamwerken voor gedistribueerde omgevingen

Object Management Architecture (OMA)

Het hart van de Object Management Architecture is Object Request Broker (ORB). De ORB is de manier waarlangs de clients en servers met elkaar communiceren. De ORB biedt standaard interfaces voor communicatie, standaard methoden voor exception handling en standaard typeren over verschillende systemen. De OMA definieert ook een verzameling begeleidende diensten (services). CORBAServices is zo een verzameling van basis diensten, men moet hierbij denken aan naming, persistentie, beveiliging, en transacties die in een geïntegreerd systeem nodig zijn. CORBADomains representeert het bedrijfsdomein en zijn gespecialiseerd in de financiële markt, productie enz. CORBAfacilities biedt diensten voor applicaties zoals internationalisatie, fout controle en afdrukdiensten. Hieronder is een afbeelding van het OMA model.



Afbeelding

Windows Distibuted interNet Architecture (DNA)

De kern van dit systeem is het Component Object Model (COM). Met dit model kunnen applicaties gebaseerd zijn op COM en de diensten die DNA biedt, zoals infrastructuur en netwerken. DNA heeft ook diensten voor onder andere systems management en beveiliging.

De DNA architectuur heeft drie lagen: de presentatie laag, applicatie laag en de data laag. Windows DNA biedt de ontwikkelaar de mogelijkheid om applicaties te ontwikkelen met presentatie op windows platformen en internet browsers. De middelste laag, de applicatie laag, bevat COM+ voor componenten diensten, Microsoft Transaction Server (MTS), MSMQ voor queuing en de Internet Information server (IIS) voor web diensten.

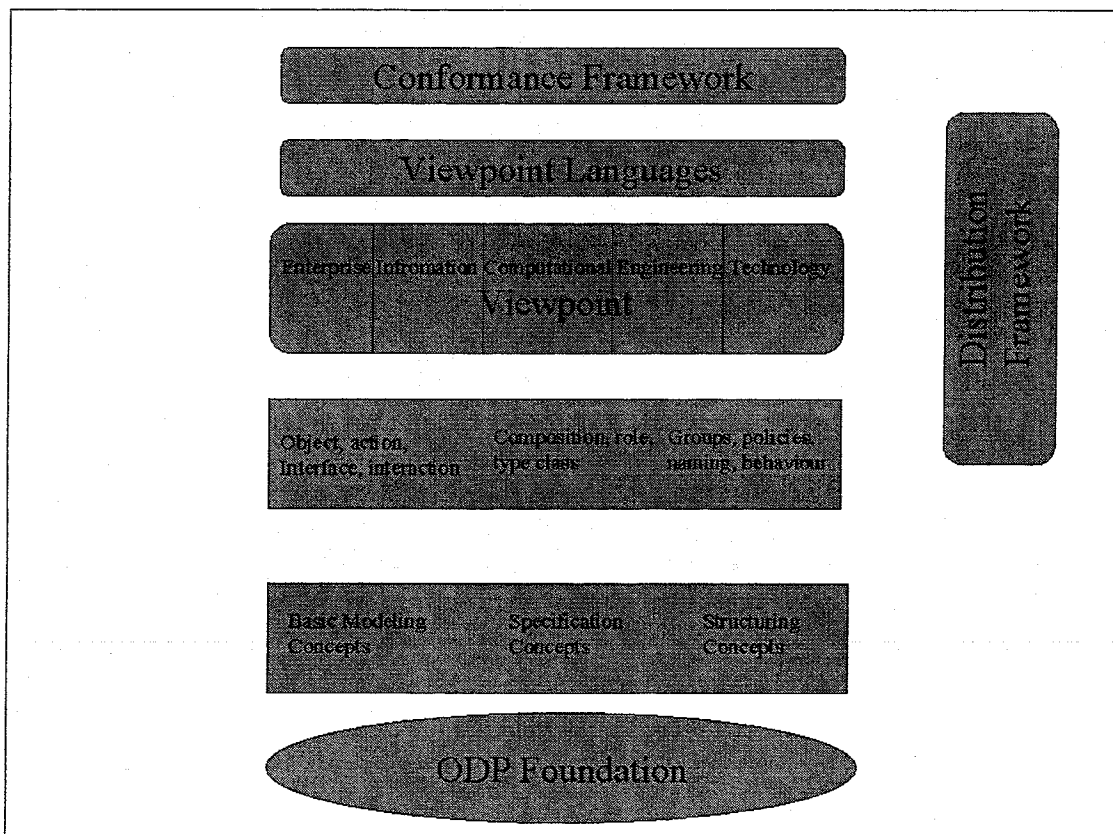
Om de derde laag te benaderen biedt DNA universele toegang tot verschillende data bronnen zoals rationele en niet rationele databases. Dit wordt gedaan via de Object Linking and Embedding Database (OLE DB) die in veel databases verwerkt zijn.

Reference Model for open distributed processing (RM-ODP)

Dit referentie model definieert de belangrijke eigenschappen van gedistribueerde systemen en definieert een raamwerk voor het ontwikkelen van open standaarden om heterogene gedistribueerde systemen te faciliteren, het bestaat uit vier delen:

- Deel 1 Een overzicht van de concepten en standaarden
- Deel 2 De fundering waar de concepten en het raamwerk gedefinieerd worden voor de beschrijving van het Open Distributed Processing (ODP) systeem, hierop kunnen viewpoints geplaatst worden.
- Deel 3 De architectuur gedefinieerd per viewpoint specificatie van een systeem en de infrastructuur die transparantie mogelijk maakt.
- Deel 4 De semantiek van de architectuur die gebruik maken van bestaande formele beschrijvingstechnieken om de model concepten te interpreteren.

RM ODP definieert vijf viewpoints die helpen gedistribueerde systeem te ontwerpen en te beschrijven: Onderneming, informatie, computational (berekeningkundig), engineering en technologie. Het technologie viewpoint beschrijft de hard- en software componenten van het systeem. Het engineering viewpoint beschrijft de interactie tussen de gedistribueerde componenten. Het computational viewpoint definieert de functionele decompositie van het systeem in objecten en hun interfaces. Het informatie viewpoint laat zien welke informatie uitgewisseld wordt en welke structuur deze heeft. Het ondernemingsviewpoint relateert het systeem aan de bedrijfseisen en bestudeert zowel de organisatie structuur alsook het doel, bereik en het beleid van het systeem.



Appendix D

Beveiliging in een gedistribueerde omgeving

Beveiliging

Enkele van de belangrijkste kwaliteitsaspecten van een bedrijfsbreed systeem zijn te scharen onder de noemer beveiliging. Bij een integratie project is het dan ook essentieel dat hier tijdig aan gedacht wordt, het liefst tijdens de ontwerp fase. Beveiliging heeft een belangrijke en positieve impact op elk niveau van het bedrijf en moet zo ontworpen worden dat de performance of schaalbaarheid ondersteund worden. Om een kwalitatieve uitspraak te kunnen doen over veiligheid moeten we eerst de doelstellingen formuleren die kunnen dan later met oplossingen.

Doelstellingen

Steeds meer bedrijven handelen via internet, het handelen via internet wordt ookwel e-commerce genoemd.

Om de behoefte aan e-commerce te bevredigen moeten bedrijven steeds opener worden, oftewel applicaties beschikbaar stellen aan derden. Om dit te kunnen realiseren moeten sterke web gebaseerde applicaties met de bestaande omgeving geïntegreerd worden. Dit brengt de nodige complicaties voor de beveiliging met zich mee. Daarom is de graad van interconnectiviteit belangrijk en verschillen de te nemen beveiligingsmaatregelen.

We onderscheiden drie groepen:

- **Internet**
Miljoenen gebruikers, gebruikers die je niet kent, decentrale controle mechanismen, moeilijk om een administratieve controle te implementeren, gevarieerde technische bekwaamheden
- **Intranet**
Duizenden gebruikers, gebruikers die bekend zijn, centrale controle, relatief makkelijk om een administratieve controle te implementeren, gestandaardiseerde technische niveaus
- **Extranet**
Samenvoeging van intranets.

Veel bedrijven hebben hun omgeving vanwege deze verschillen beveiligd met firewalls. Een vaak gemaakte fout is te denken dat door het beveiligen van een omgeving met logins en wachtwoorden het binnen deze omgeving ook veilig is. De toenemende complexiteit van de interconnectiviteit heeft de beveiligde omgeving aanpak moeilijker gemaakt. De notie van extranets bijvoorbeeld compliceert dit eenvoudige model al.

Een bedrijf wil een partner niet volledig bezit geven van zijn eigen systeem omdat dat andere bedrijf er misschien een andere veiligheidsbeleid op na houdt.

Om informatie bronnen te beschermen, moeten geautoriseerde gebruikers toegelaten worden terwijl anderen geweigerd worden. Alhoewel de technieken regelmatig veranderen kan men beveiliging samenvatten in vier hoofddoelen:

- **Vertrouwelijkheid**
Ook wel confidentialiteit. Vertrouwelijkheidsbeveiliging zorgt er voor dat informatie alleen toegankelijk is voor die genen waarvoor het bedoeld is. De privacy van de gebruikers wordt gewaarborgd en de diefstal van informatie wordt voorkomen. Dit wordt gedaan met informatie die verstuurd wordt en met opgeslagen data
- **Integriteit**
Integriteit wil zeggen dat informatie alleen aangepast kan worden door gebruikers die daar recht toe hebben en dat deze op een correcte manier doen. Het houdt ook in dat data nergens op het pad getransformeerd kan worden.
- **Aansprakelijkheid**
Aansprakelijkheid betekent dat gebruikers verantwoordelijk gemaakt kunnen worden voor hun acties. Deze beveiliging wordt gebruikt om aanvallen te traceren naar personen.
- **Beschikbaarheid**
Dit deel van de beveiliging moet er voor zorgen dat het systeem continu beschikbaar is. Het systeem mag niet door kwaadwillende plat gelegd worden, noch door fouten.

Deze doelen kunnen worden bereikt indien aan volgende eisen voldaan is:

- **Identificatie en authenticatie**
Dit is het proces van het verifiëren van de identiteit van een gebruiker. Hier aan verbonden is de toegangscontrole. Authenticatie gebeurt meestal op basis van iets wat de gebruiker heeft (wachtwoord), iets wat hij heeft (een token) of iets persoonlijk (vingerafdruk) . Ook combinaties worden gebruikt om de veiligheid te verhogen.
- **Toegangscontrole**
De toegangscontrole is het proces van het beperken van toegang naar beschermde bronnen tot enkel geautoriseerde gebruikers. Deze toegang kan beperkt zijn voor bepaalde gebruikers, in een veiligheidsbeleid, locatie gebonden of functie gebonden. De toegangscontrole speelt dus een rol in zowel de confidentialiteit alsook de integriteit
- **Beveiliging van de communicatie**
Hiermee wordt de beveiliging van de data stroom bedoeld via de lagere communicatie lagen, de transport laag bijvoorbeeld, meestal wordt dit gedaan via encryptie.
- **Non repudiation**
Hier worden de niet tegensprekbare bewijzen voor (trans)acties vastgelegd. Het moet mogelijk gemaakt worden voor de ontvanger om te zien waar de data vandaan komt, of de ontvangst van data door de verstuurder, om te voorkomen dat mensen het versturen of ontvangen van berichten ontkennen.
- **Veiligheidsaudits**
Dit soort audits worden gebruikt om het veiligheidsbeleid te controleren. Veiligheid gerelateerde acties worden opgeslagen door het systeem. Gebruikers worden verantwoordelijk gehouden voor deze acties.
- **Beveiligingsadministratie**
Hier worden de data en metadata beheerd die nodig zijn om een bepaald veiligheidsbeleid te voeren.

Risk management

Om de veiligheid binnen een bedrijfsbreed systeem te kunnen beoordelen moeten we verder kijken dan de applicaties alleen. De veiligheidsgraad binnen een organisatie is van meer zaken afhankelijk dan enkel technologie, zo zijn de regels en mechanismen om ze op te leggen essentieel. In deze regels wordt het veiligheidsbeleid vast gelegd. Het veiligheidsbeleid is een verklaring waarin staat wat de veiligheidsdoelen van een organisatie zijn en welke regels er gehanteerd worden. Meestal wordt dit document vergezeld met een precieze beschrijving van de veiligheidsprocedures, zo dat elke gebruiker weet wat van hem verwacht wordt. Problemen ontstaan er vaak bij het installeren van nieuwe software of veranderingen in configuraties. Het lijkt dan ook essentieel dat een E.A.I. oplossing de verantwoordelijkheid voor een groot deel van de veiligheidsdoelstellingen kan dragen. Door de centrale positie kunnen een deel van de technische veiligheidseisen vervuld worden, het beleid echter niet. Hiervoor moet scholing, controle en het monitoren van de activiteiten van gebruikers een oplossing bieden. Het proces van risk management biedt momenteel de beste oplossing voor organisaties die zich een weg willen banen tussen de keuzes binnen het gebied van beveiliging.

Risk management is het afwegen van kansen tegen risico's. Bij deze aanpak worden risico's niet uitgesloten maar worden deze onder controle gehouden.

Het doel is het adequaat genoeg beveiligen om de doelstellingen te halen zonder onnodige risico's te nemen. Men moet de goede afweging maken tussen veiligheid, kosten, performance en functionaliteit. De risico's van een bepaalde technologie worden afgewogen tegen de mogelijke voordelen. Door de drastische verhoging in connectiviteit van de netwerken hebben veel organisaties te maken met inbrekers, die bijvoorbeeld via het internet proberen een poging wagen. Eenmaal binnen kunnen deze hackers informatie veranderen, frauderen, saboteren, spioneren enzovoorts.

Deze risico's moeten afgewogen worden tegen de kansen. In de meeste gevallen zijn de baten van een automatisering, het elektronisch uitwisselen van data en internetworking hoger dan de risico's in geval van veiligheidsproblemen. Er is een grijs gebied tussen het helemaal niet gebruik maken van net-applicaties en het te snel openstellen van een onbeveiligd systeem of het tot in extremis beveiligen van een applicatie zodanig dat er niet meer mee te werken valt. Zo een beslissing is afhankelijk van de doelstellingen een bankier zal er veel aangelegen zijn dat transacties goed gedocumenteerd worden en dat er naderhand geen discussie over de identiteit van de bron is, een ISP zal veel waarde hechten aan

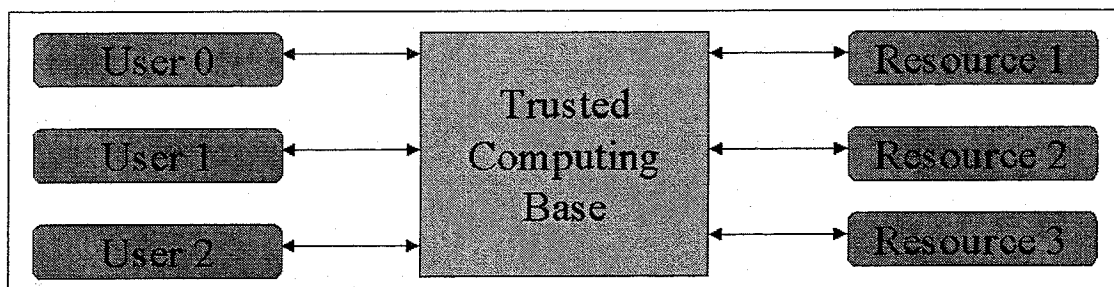
de performance. In een ziekenhuis is de data integriteit belangrijk omdat geen onbevoegden data over een patiënt mogen veranderen, een militair systeem moet confidentieel zijn zodat er geen geheimen kunnen worden gestolen.

De uitdaging voor allemaal is de beveiliging zo te implementeren dat zijn/haar veiligheidsdoelstellingen zo kosteneffectief mogelijk bereikt worden. Om een beslissing te kunnen maken tussen risico's en baten moet men een bedrijfsbrede visie hebben. Met de komst van de gedistribueerde middleware en gedistribueerde applicaties ontstond er een nieuw veiligheidsprobleem.

Gedistribueerde veiligheid

Object georiënteerde technieken waarin data en logica gegroepeerd worden heeft een dramatische impact gehad op het systeem landschap.

Gedistribueerde object componenten bevatten code en data en kunnen zich overal op het netwerk bevinden. Door de hoge mate van doorzichtigheid en hergebruik van componenten biedt deze techniek vele mogelijkheden, maar er dienen zich veel complexere veiligheidsvraagstukken aan. Er moet dan ook anders gedacht worden en er moeten andere tools ontwikkeld worden. De eisen zijn niet veranderd ze moeten echter anders gewaarborgd worden in een gedistribueerde omgeving. In systemen met een mainframe met data centres en professioneel personeel was de beveiliging eenvoudig te realiseren. Een verzameling veiligheidsmechanismen werden centraal in een trusted computing base (TCB) geïmplementeerd. De TCB was klein genoeg om degelijk geanalyseerd te worden om te kunnen garanderen dat de veiligheidseisen gehaald werden. Zoals in afbeelding 1 te zien is werkt de TCB tussen de gebruikers en de bronnen.

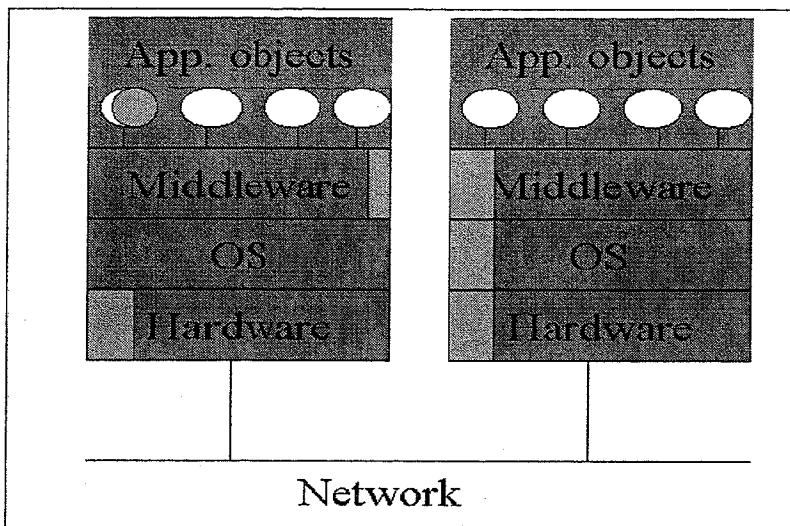


Afbeelding 1

De TCB beschermt de bronnen door eerst te controleren of een gebruiker rechten heeft en hem daarna toegang te geven. Andere functionaliteiten zoals bijvoorbeeld het auditten of het monitoren kunnen ook in deze component geïmplementeerd worden.

In een gedistribueerde omgeving is de veiligheidsfunctionaliteit gedistribueerd door de applicatie architectuur. Fysieke veiligheid kan niet meer worden gegarandeerd en nieuwe gevaren bieden zich aan. De verantwoordelijkheid voor de veiligheid kan zich dan ook in de applicaties zelf, door de middleware, door het operating systeem, door de hardware of door een combinatie van de genoemden gerealiseerd worden. Omdat vele nieuwe gedistribueerde object interactie hebben met legacy of bestaande hardware moet de veiligheid ook voor de koppelingen gewaarborgd kunnen worden. Dit resulteert in een complexere veiligheidsarchitectuur. De centrale oplossing biedt geen mogelijkheden maar er moet een aanpak gekozen worden waar mee ook gedistribueerde data beveiligd kan worden. Een voorbeeld van een veiligheidsarchitectuur is in afbeelding X te vinden.

De grijze stukken geven aan waar de beveiliging geregeld wordt, in het linker systeem bijvoorbeeld is deze gedeeltelijk in de hardware, middleware en in de applicaties zelf verzorgd. Het distribueren van veiligheid betekent dat een bepaalde gedistribueerde applicatie veilig is, maar dat het lastig is om dat aan te tonen. Met het toenemen van het aantal platforms en variëteit in toegang neemt de moeilijkheid van het analyseren van de veiligheid exponentieel toe (Ron Zahavi). De factoren die het analyseren van gedistribueerde object systeem veiligheid compliceren zijn:



Afbeelding 2

- Gelaagde architectuur. Men moet naar alle lagen kijken en niet naar de implementatie in 1 laag.
- Data beschikbaar op het netwerk. In gedistribueerde object systemen is het mogelijk om via internet of grote intranetten te werken, dit betekent dat de data gevoelig is voor packet-sniffing.
- De dynamische systeem compositie. In een object systeem kunnen 'on the fly' objecten aangemaakt worden. Dit betekent dat de veiligheid dus ook dynamisch moet werken
- Multienterprise. Bij het integreren van meerdere bedrijven botst het verschil in veiligheidsbeleid soms.
- Complexe veiligheid administratie. De flexibiliteit en openheid van een systeem zorgen ervoor dat de veiligheidsadministratie een complexe zaak wordt.

Gebruikte beveiligingstechnieken

Om aan de veiligheidseisen te kunnen voldoen zijn meerdere technieken ontwikkeld. De meeste zijn gebaseerd op cryptografische methoden. We zullen nu in vogelvlucht enkele technieken beschrijven die in een E.A.I. omgeving er voor kunnen zorgen dat aan de veiligheidseisen voldaan wordt.

Geheime en publieke sleutel cryptografie.

In de geheime sleutel cryptografie, ook wel symmetrisch, wordt dezelfde sleutel gebruikt voor het encoderen en het decoderen. De sleutel moeten aan de gebruikers aangeboden worden op zo een manier dat naderen er niet aan kunnen, bijvoorbeeld met een koerier. Na verdeling van de sleutels moeten deze beheerd worden wat nogal al wat organisatie vergt. Geheime sleutel cryptografie werkt sneller dan publieke sleutel cryptografie. Bij publieke sleutel cryptografie heeft elke gebruiker twee sleutels: een publieke en een geheime. Zoals de naam doet vermoeden wordt de publieke sleutel openbaar gemaakt en heeft elke gebruiker zijn eigen geheime sleutel. De zender en de ontvanger hoeven niet de geheime informatie te delen, maar alleen de publieke sleutel is nodig om te communiceren. De zender gebruikt de publieke sleutel van de ontvanger om het bericht te coderen, wat deze weer met zijn geheime sleutel kan decoderen.

Dus zelfs de zender kan het bericht niet meer decoderen nadat hij het versleuteld heeft.

Publieke sleutel systemen worden gebruikt om non repudiatie via digitale handtekeningen te garanderen.

Secure Socket layer protocol

Dit protocol werd ontwikkeld door netscape en wordt gebruikt om veilig via het internet te communiceren.

SSL biedt:

- Wederzijdse authenticatie tussen de communicerende entiteiten
- Integriteit van het data transport
- Bericht versleuteling

SSL biedt de mogelijkheid andere protocollen erboven te gebruiken. SSL is een zogenaamd handshake protocol. Eerst bewijst de server zijn identiteit aan de client (server side authentication), en soms moet ook de client zijn identiteit bewijzen. Deze handshake is gebaseerd op publieke sleutel cryptografie.

Simple Public key Mechanism

Simple Public key Mechanism (SPKM) is een publiek sleutel protocol van Entrust Technologies. SPKM ondersteunt identiteit gebaseerde cryptografie, maar geen beleid gebaseerd op rollen of groepen. SPKM gebruikt het mechanisme om digitale handtekeningen te kunnen maken en zo de identiteit van de bericht zender te achterhalen.

Kerberos

Kerberos is een authenticatie protocol dat ontwikkeld werd op het Massachusetts Institute of Technology (MIT). Kerberos werd ontwikkeld in verband met het probleem van het versturen van niet gecodeerde wachtwoorden via netwerken, zoals bijvoorbeeld gebeurd als men een telnet sessie in plaats van een ssh sessie opent. Kerberos gebruikt geheime sleutel cryptografie voor:

- Client en server authenticatie
- Data encryptie

In zo een systeem wordt een aangewezen server gebruikt voor het toewijzen en beheren van sleutels.

Distributed computing environment (DCE)

Deze oplossing is speciaal voor gedistribueerde omgevingen gemaakt door de Open Software Foundation (OSF). DCE moet volledige veiligheid beiden in een gedistribueerd systeem. Er wordt gebruik gemaakt van Kerberos als authenticatie protocol en maakt gebruik van een 'security-server' waarop gebruikers informatie wordt opgeslagen. De gebruikersrechten worden vastgelegd in zogenaamde 'privelege attribute certificates' (PACs).

DCE biedt in tegenstelling tot de andere beschreven technieken ook toegangscontrole. Er wordt naar aanleiding van een lijst toegang verleent tot objecten aan bepaalde gebruikers.

Secure European System for Applications in a multivendor Environment (SESAME)

SESAME is een Europees R&D project, waaruit een aantal gereedschappen voor het ontwikkelen uit voortvloeiden:

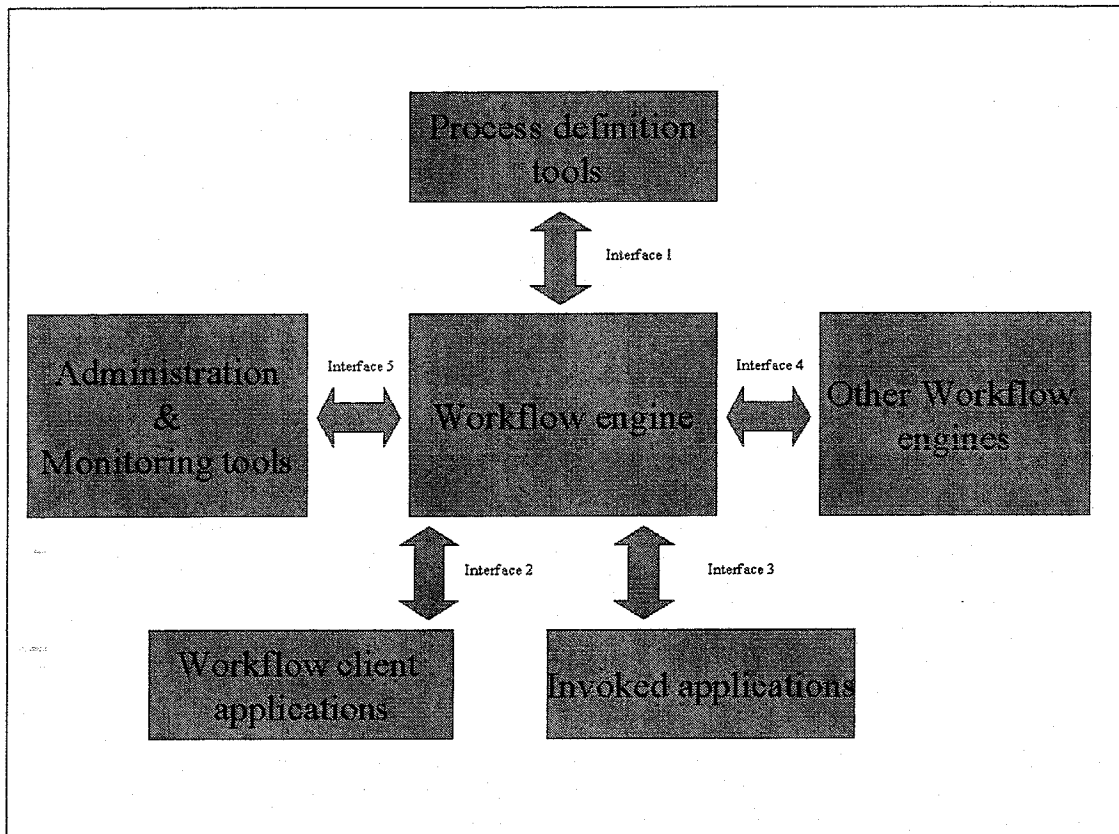
- Single sign on
- Op rollen gebaseerde gedistribueerde toegang door gebruik te maken van PACs
- Cryptografische bescherming van uitwisselingen tussen gebruikers en applicaties
- Schaalbaarheid door gebruik te maken van publieke sleutel technologieën

Aspect	Efficiëntie	Effektiviteit	Educatief	Flexibiliteit	Gebruikersvriendelijkheid	Compatibiliteit	Beveiligbaarheid
Methodologische principes							
Lering en abstractie					x	x	x
Hergebruik			x				x
Interoperabiliteit			x				x
Samenwerken en Pakketten			x			x	x
Beveiligbaarheid							
Veiligheid					x	x	x
Parasiteerbaarheid			x	x	x		
Life cycle					x	x	x
Herengedragelijkheid			x	x			x
Identificeer Objecten				x	x		x
overdracht	x	x					
Compatibiliteit en interoperabiliteit							
Interactie				x	x		x
Data formaten				x			x
Directe interactie				x			x
Interoperabiliteit			x	x			
Compatibiliteit							
Ontwikkelbaarheid			x				x
Zelfde beheer						x	x
Testbaarheid en foutopsporing			x			x	x
Compatibiliteit							x
Compatibiliteit							
Compatibiliteit						x	x

Appendix F Workflow management

1 Workflow raamwerk

Omdat het workflow modelling tool een kern van de EAI applicatie zal worden zullen we nu nader ingaan op workflow management en een raamwerk beschrijven voor workflowmanagement. We zullen zien dat op basis van het raamwerk voor workflowmanagement en de eisen aan adaptoren een generieke componenten architectuur voor de ideale E.A.I.-oplossing kan worden afgeleid. De WEMC heeft een raamwerk voor de definitie van workflowstandaards voorgesteld. Dit raamwerk bevat vijf interface standaards voor samenwerking en communicatie met behulp waarvan meerdere workflow-management producten of componenten kunnen samenwerken in een gebruikers omgeving



Van de vijf categorieën zijn op dit moment interface twee en vier gedefinieerd en interface één in concept beschikbaar. We gaan nu een nadere beschrijving van de basis functionaliteiten van een workflow-managementsysteem geven. Deze werden ingedeeld in volgende zes groepen:

- Procesdefinitie
- Administratie
- Organisatiedefinitie
- Procesmonitor
- Taaklijst
- Workflowengine

Hiermee worden de workflowmanagement producten uit afbeelding X gedekt. Deze worden hieronder nader beschreven.

Procesdefinitie

De standaardfunctionaliteit van een workflow tool is een applicatie voor procesdefinitie. Vrijwel elk product heeft een grafische omgeving, waarbinnen een proces is te modelleren in termen van activiteiten en de mogelijke stroom van taken langs deze activiteiten.

Doelstelling van dergelijke grafische hulpmiddelen is de vertaalslag van een procesbeschrijving naar het desbetreffende workflow-management systeem zo gebruiksvriendelijk te maken. Binnen veel procesdefinitietools kan op uitgebreide manier de grafische weergave met teksten commentaar worden uitgebreid. De modelleer technieken lijken over het algemeen op hiërarchische klassieke petri-netten, maar meestal zijn ze niet volledig conform. Dit is jammer als men denkt aan de analyse-mogelijkheden die wiskundig gedefinieerde modelleertechnieken. Een belangrijk kenmerk van een workflow-managementoplossing is dat deze raakvlakken heeft, of zal krijgen met vrijwel elk geautomatiseerd systeem in een organisatie. Vandaar ook de nauwe verwantschap met E.A.I. oplossingen die naast het integreren van applicaties ook de coördinatie tussen de applicatie in de hand willen houden. Integratie van bestaande systemen en databases is op dit moment de grootste kostenpost in een workflow implementatie traject en dit zal in de toekomst naar verwachting niet veranderen, hierdoor ontstaat er een opportunity voor verkopers van E.A.I oplossingen gezien de geringe marginale verschillen in prijs ten opzicht van WfMC oplossingen.

Administratie

De administratie is het onderdeel van een workflowmanagementsysteem waar zaken worden vastgelegd, zoals gebruiker en bijbehorende rollen en rechten, definitie van de workflow relevante gegevens en eventuele formulieren waarop informatie betreffende een taak worden verzameld.

Organisatiedefinitie

Het modelleren van de organisatie is een functionaliteit die door veel workflowproducten niet of nauwelijks wordt ondersteund. Dit in tegenstelling tot E.A.I. producten waar ze meer in trek zijn. De functionaliteit is vaak beperkt tot de hiërarchische organisatiestructuur, voor zover van belang voor het toekennen van de rollen.

Procesmonitor

Dit onderdeel ook wel dashboard genoemd geeft inzicht in de voortgang van processen. Het management en degenen die voor het proces verantwoordelijk is in staat actueel inzicht te verkrijgen in de status van alle lopende processen en de werkvoorraad. Ook zijn overzichten met kengetallen eenvoudig te maken zoals gemiddelde doorlooptijden en gerealiseerde aantallen.

Taaklijst

Met de functionaliteit 'taaklijst' wordt de methode bedoeld waarop de taken aan de diverse gebruikers wordt aangeboden. Veel huidige workflowproducten beschikken over een eigen Graphical user interface. Vaak worden de taken aangeboden via de mailinterface. Hierbij wordt door het workflow-managementsysteem gebruik gemaakt van de user interface van de gebruikte mailapplicatie. Een nadeel is dat deze taken mits daar een persoon voor nodig is asynchroon aangeboden worden.

Workflow engine

Het centrale onderdeel van elk workflowmanagementsysteem is de workflow engine. Dit is het deel dat tijdens uitvoer het procesmodel interpreteert en de taken toewijst aan de diverse gebruiker volgens de in het model vastgestelde regels.

Appendix H Case studies

Case study
VITRIA BusinessWare

Bedrijf : Vitria Technologies
Product : BusinessWare
Website : www.vitria.com
PWC-Partner : Ja

In deze case study wordt het afgeleide model getoetst aan de praktijk. De eerste EAI. aanbieder die onder de loep genomen wordt is Vitria. Het toekennen van componenten aan het model heeft plaatsgevonden aan de hand van de gegevens die Vitria over hun eigen product ter beschikking stelt in de whitepapers, de technische specificatie en de ingevulde vragenlijst.

Vitria biedt een complete EAI oplossing namens BusinessWare; deze is volledig geprogrammeerd in een Java omgeving. Alle operaties, inclusief ontwikkeling en administratie, worden uitgevoerd in een grafische console die gebruik maakt van de Java AWT klassen. BusinessWare is geprogrammeerd volgens de CORBA standaard en kan dus getoetst worden aan de eerder beschreven architectuureisen voor gedistribueerde systemen (Zie Appendix E). Berichten worden als *remote procedure calls* over IIOP behandeld. In Vitria jargon wordt aan berichten gerefereerd als *events*. Een andere product van Vitria is de *Value Chain Markup Language* (VCML) waarmee de B2B integratie geregeld wordt, hier wordt later dieper op ingegaan.

Het product bestaat uit vier hoofdcomponenten:

Automator Een *business process modeling tool* (BPM)
Communicator Publish/subscribe bericht systeem
Connectors Adaptoren voor de overige applicaties
Analyzer real time statistieken en een *dashboard* (grafisch weergave van events)

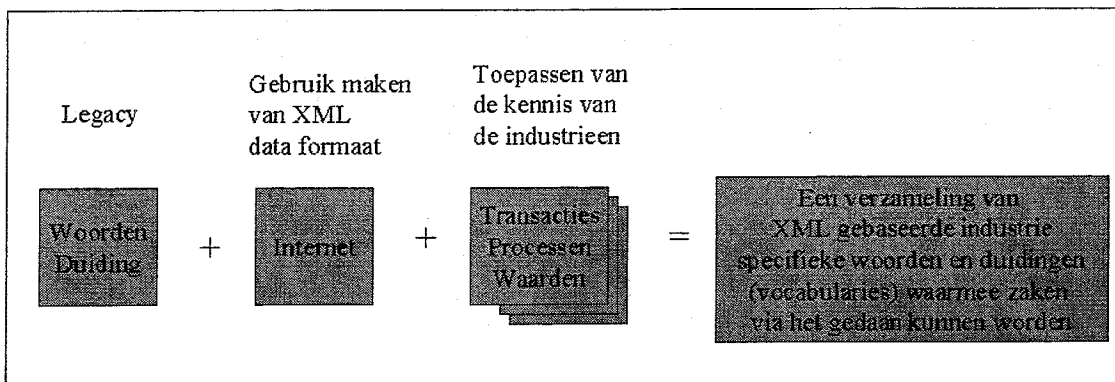
VCML

Om het mogelijk te maken voor externen om met VITRIA BusinessWare te communiceren is er een 'mark up'-taal ontwikkeld, VCML. VCML is een afkorting voor *Value Chain Markup Language* is een XML dialect en is door VITRIA tot standaard uitgeroepen. Dit omdat er, naar eigen zeggen, nog geen door de industrie geaccepteerde XML standaard is. Er zijn al wel XML standaarden voor specifieke industrieën, maar men is bang dat het ontwikkelen van een algemene standaard net zoals bij EDI (electronic data interchange) nog jaren kan duren.

VCML is het eerste initiatief dat tracht bestaande b2b documenten in een XML formaat aan te leveren, zodanig dat de structuur en de in de bestaande document formaten zoals EDI verwerkte *business rules* te behouden.

Men kan door gebruik te maken van de gestandaardiseerde *dictionaries* eenvoudig een conversie maken tussen bestaande documenten en XML. Op dit moment biedt Vitria een verzameling van 3100 van de meest gebruikte EDI en XML standaarden

Concluderend kan men de volgende vergelijking opstellen:

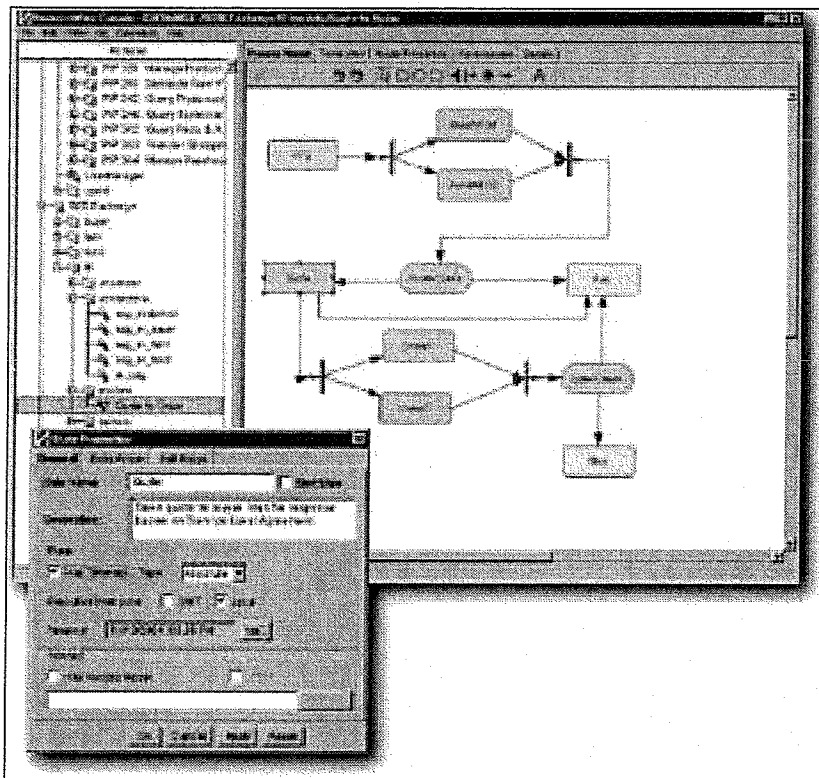


Automator

De automator is de kern van het workflowmanagement systeem en procesbeheersysteem in Vitria, welke is gemaakt om grote volumes van transacties te verwerken. De Automator heeft een grafische interface om processen te modelleren.

Het modelleren gebeurt naar de *Unified modelling language* (UML) [REF XY]en door de Workflow Management Coalition (WfMC) geaccepteerde standaarden. Er worden geavanceerde modelleertechnieken geboden zoals het nesten van modellen, geparametriseerde modellen en de mogelijkheid dynamisch submodellen te invoeren. Men kan modellen van andere partijen importeren met behoud van data-integriteit importeren.

Het uitvoeren van de processen gebeurt direct, dit betekent dat men het proces kan aanpassen zonder het systeem te onderbreken. In een netwerk staat in de *directory* waar bronnen gevonden kunnen worden. De automator maak gebruik van het *Lightweight Directory Access Protocol* om van de directory diensten gebruik te maken. De automator zorgt ervoor dat toestanden persistent zijn, dit betekent dat in het geval van excepties gemiste stappen gecompenseerd kunnen worden.



Communicator

De communicator biedt een robuust bericht systeem voor BusinessWare. Communicator maakt gebruik van de *publish en subscribe* methode die in het hoofdstuk 8.2 beschreven wordt. Naast communicatie biedt Communicator functies zoals *Quality of Service* (QOS), replicatie en transactionaliteit. Intern maak de Communicator gebruik van CORBA om berichten te versturen. De transactionaliteit wordt geboden door gebruik te maken van CORBA's Transaction Service en voor de naamgevingsstrategie wordt gebruik gemaakt van de CORBA Naming Service, deze koppelt een naam aan een *interoperable object reference* (IOR) waardoor het niet uitmaakt als objecten bijvoorbeeld verplaatst worden.

Connectors

De connectoren voor BusinessWare bieden de mogelijkheid om systemen aan businessWare te koppelen. Connectoren herkennen automatisch veranderingen, events en converteren de informatie in het gewenste formaat. De ConnectorsKit maakt het mogelijk grafisch zelf connectoren te definiëren. De connectoren die standaard met businessWare geleverd worden zijn de volgende:

Clarify (CRM)
CommerceOne
CORBA
Daleen (telco billing)
DB2
Email
Flat Files
HTTP/HTTPS
Informix
JDBC
Kenan Arbor (telco billing)
MetaSolv (telco)
MQSeries
MS SQL Server
ODBC
Oracle Applications
Oracle Database
Portal Infranet (telco billing)
Remedy
RosettaNet
SAP R/3 (BAPI) *
SAP R/3 (IDOC) *
Siebel (CRM)
Single.view (aka Saville)
Sybase
Syndesis (telco)
XML

Analyzer

De Analyzer biedt de mogelijkheid snel sleutelstatistieken te bekijken en te waarschuwen als bepaalde doelstellingen niet gehaald worden of juist wel. Daarnaast biedt de Analyzer *real time* grafisch inzicht in de uitvoering van het proces. Men kan via het *dashboard* processen definiëren, analyseren en optimaliseren. Deze cockpit biedt zijn informatie aan via het net; dit betekent dat men de informatie overal waar internet beschikbaar is kan bekijken en eventueel kan ingrijpen.

Overige diensten

De gedistribueerde architectuur biedt *scalability*, transparante administratie en configuratie van fysiek gescheiden systemen.

Beveiliging: een op SSL gebaseerd raamwerk voor authenticatie, toegangscontrole, encryptie en een *Access Control list*.

Systeem beheer: compatibel met SNMP om samen te kunnen werken met bekende systeembeheer-software zoals Open View of Tivoli

Metadata beheer: biedt de optie om opgeslagen modellen, business objects, parameter configuraties en processen te hergebruiken

Verder: Load balancing, opties voor hogere beschikbaarheid, routing, audit en logging

Het Model

De hierboven beschreven componenten passen op volgende manier in het model:

<p>Inter-Enterprise Process Automation Ondersteund HTTP, HTTPS en E-Mail/FTP. Momenteel wordt er aan een de Transformatie aspect met een XML component</p>	<p>Application Services Communication Services Metadata Services Operational Services</p>	<p>Adaptive Enterprise CloudReady Enterprise CloudReady Enterprise CloudReady Enterprise</p>
<p>Business Process Automation Administratief en niet een volledig BPM tool. Het is de mogelijkheid om de status te bepalen. Een artikel heeft de mogelijkheid om te worden</p>		
<p>Integration Tools Administratief en niet een volledig BPM tool. Het is de mogelijkheid om de status te bepalen. Een artikel heeft de mogelijkheid om te worden</p>		
<p>Application Content Information Java Programmable. Dit raakt niet wordt gebruikt om code in te ontwikkelen om de verspreiding te maken. Tevens is er een grafische tool die gebruikt wordt in andere gevallen.</p>		
<p>Communication Services Communicatie. Dit is onder andere de levering van berichten. Er wordt gewerkt aan IDL.</p>		
<p>Infrastructure Standards Windows NT, Unix, Solaris, HP-UX, OpenVMS, TOPP en HOP.</p>		
<p>Metadata Services Console. Dit is de bestaande vorm van de metadata op de server van <i>OpenPaaS</i> database.</p>		
<p>Operational Services Console. Administratie kan via een GUI. Automatische Java wordt in <i>Oracle Metadata</i> opgeladen (Backup). Message queue worden in deze fase niet opgeladen (in de volgende backup). Maak gebruik van de <i>Parasoft</i> netwerk monitor. Het is belangrijk van de backup te maken.</p>		

Case Study

Bedrijf : Crossworlds
Product :
Web site : www.crossworlds.com
PWC partner : Ja

Inleiding

In dit document wordt het product van de EAI aanbieder Crossworlds geanalyseerd. Het afbeelden van de componenten is gebeurd aan de hand van de door Crossworlds beschikbaar gestelde *whitepapers* en de door Crossworlds ingevulde PwC vragenlijst.

Crossworlds biedt een volledige integratie-oplossing die bestaat uit vele deeloplossingen. Deze deeloplossingen kunnen ook onafhankelijk van elkaar functioneren. Zo kunnen bedrijven langzaam maar zeker naar een volledig geïntegreerd bedrijfsbreed systeem toe groeien. De onderliggende architectuur is 'hub & spoke' waarbij de integratie producten in de hub verwerkt zijn.

De producten zijn in drie categorieën ingedeeld: infrastructuurproducten, applicatie-integratie producten en proces-integratie producten.

De kern van het systeem is een server van Crossworlds: de InterChange Server; deze is gemaakt in Java en men legt de nadruk op performance en schaalbaarheid. Een voordeel van deze server is dat gebruikers een centrale gemeenschappelijke *view* hebben over bedrijfsbrede data. Daarnaast wordt voor de infrastructuur een systeembeheerpakket geboden, waarmee het mogelijk is om te controleren en monitoren wat er gebeurt .

De Server werkt voor synchrone communicatie volgens CORBA met IIOP en voor de asynchrone communicatie wordt gebruik gemaakt van MQ Series, een *message queuing systeem* van IBM.

Naast de server biedt Crossworlds meerdere applicatie integratie producten.

Er wordt bijvoorbeeld een onderscheid gemaakt tussen applicatie-adaptoren en technologie-adaptoren; met deze laatste worden adaptoren voor ODBC, JDBC, HTML, CORBA, XML, EDI etcetera bedoeld.

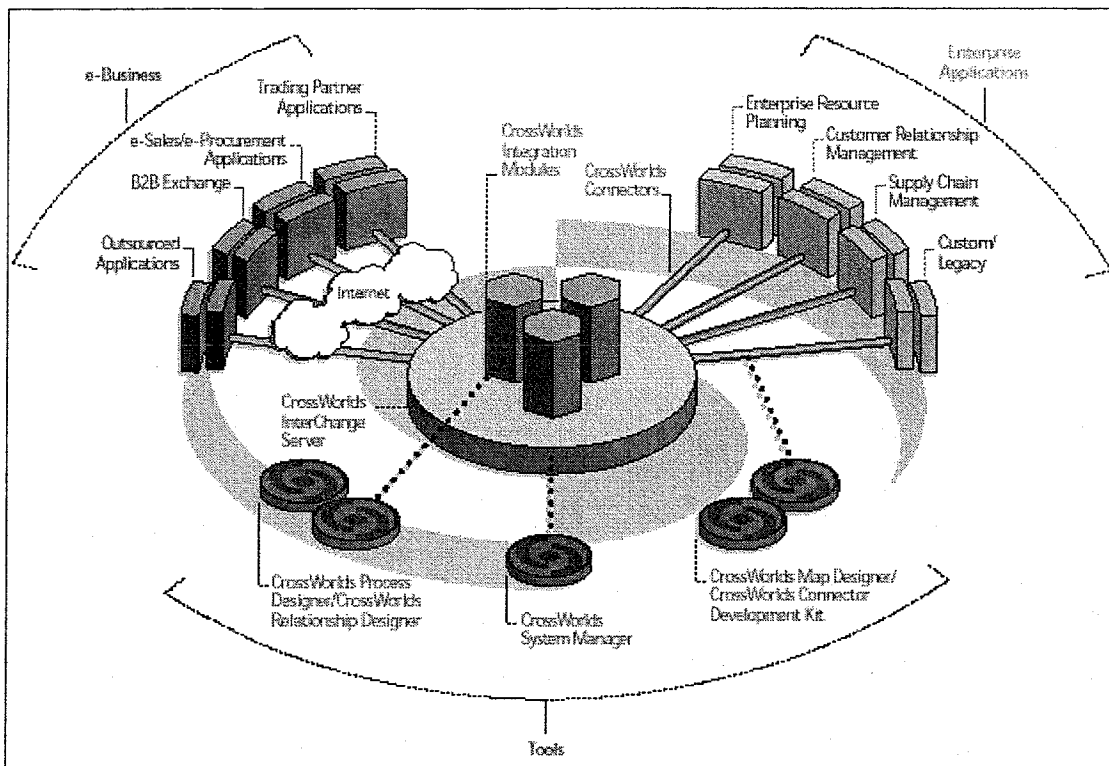
Daarnaast is er ook een product beschikbaar waarmee het mogelijk is om zelf transformaties en connectoren te ontwikkelen.

De derde groep zijn de componenten ter ondersteuning van de procesintegratie. Er wordt een tool geboden waarmee men met een grafische interface processen kan definiëren. Daarnaast biedt Crossworlds een verzameling voor gedefinieerde processen voor bijvoorbeeld e-commerce of interactie met leveranciers. Deze zijn geoptimaliseerd qua performance maar hebben als nadeel dat het bedrijfsproces dus aan het voorgedefinieerde proces dient worden aangepast.

De laatste component uit deze categorie wordt gebruikt om een data-isolatielaag aan te brengen.

Hiermee wordt de flexibiliteit en de onderhoudbaarheid verhoogt omdat het een intern datamodel is.

Het hele systeem ziet er dan als volgt uit:



De componenten worden in zes diensten ingedeeld in Crossworlds:

- Interchange server
- Collaborations
- Trading Partner Interchange
- Connectors
- Business objects
- Server access interface

Interchange Server

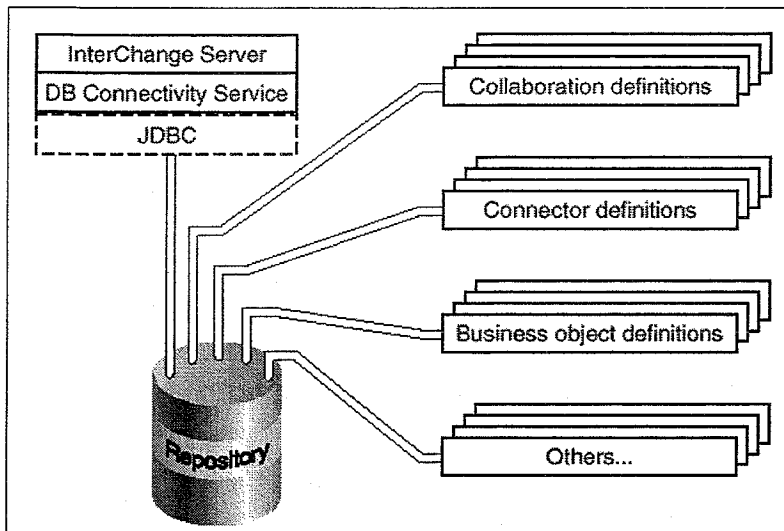
De Interchange server biedt een rijke verzameling diensten voor gedistribueerde systemen. Deze diensten zijn gebaseerd op het CORBA model. De server is *multithreaded*, *Java-based* en *event driven*. Met een event wordt in de Crossworlds omgeving een verandering in de toestand van de data van een gekoppelde applicatie bedoeld.

Een aantal diensten die geboden worden door de server zijn in volgende groepen in te delen:

Stelsel beheer

Event management: de objecten die tijdens een proces aangeroepen worden, worden persistent opgeslagen. De server kan dus na een fout snel de vorige toestand herstellen. Verder is er een *failure component* die het mogelijk maakt te loggen en bieden mislukte events opnieuw aan.

Database verbindingen: oftewel *database connectivity service*. De Interchange Server heeft een zogenaamde *repository* waar configuraties en definities persistent worden opgeslagen. Dit systeem managed de verbindingen tussen de server en het *repository*. De onderstaande afbeelding verduidelijkt dit.



Routing: De business objects moeten via connectoren en *collaboration* (zie volgende paragraaf) door het systeem gestuurd worden, deze dienst zorgt ervoor dat dit op een zinvolle manier gebeurt. Er wordt gebruik gemaakt van een messaging systeem van IBM: MQSeries en een Object request broker van Visigenics (CORBA geïnspireerd)

Transacties: Deze dienst zorgt ervoor dat processen uitgevoerd worden met de zogenaamde ACID eigenschappen. ACID staat voor:

- Atomicity
- Consistency
- Isolation
- Durability

Om deze transactionele consistentie te waarborgen is het altijd mogelijk om terug te rollen per processtap. Men maakt gebruik van de *two phase commit* strategie. Deze houdt in dat een stap oftewel volledig uitgevoerd wordt of terug gedraaid.

Mapping: Voor het mappen wordt of een product van derden ,de TSI Mercator, gebruikt of de mapper van Crossworlds.

Relaties: Er wordt gebruikt gemaakt van tabellen waarin de relaties tussen sleutel waarden van de verschillende applicaties in op worden geslagen. Dit wordt gedaan om het probleem van de verschillende en niet compatibele index schema's van applicaties op te lossen.

Collaborations

In een *collaboration* worden binnen het Crossworlds systeem de bedrijfsprocessen opgeslagen. Het helderst kan dit aan een voorbeeld van een van de standaard meegeleverde *collaboration* getoond worden. De *collaboration* ContractSync bevat de logica om verschillende applicaties te koppelen en data tussen hen uit te wisselen door het linken van de functies ervan. De *collaboration* draaien op de interchange server en elke *Collaboration* kan interacteren met een aantal applicaties via de connectoren. Met behulp van de *Process designer* kan men zelf *collaborations* maken.

Trading Partner Interchange

Deze component verzorgt de integratie met externe applicaties. Er zijn twee mogelijkheden: of de partner heeft ook Crossworlds software of niet. In het eerste geval kan men terug vallen op *collaborations*, zo niet dan komt de TPI om de hoek kijken. Het is een open standaard gemaakt voor 'losjes gekoppelde' (*'loosly coupled'*) handelsrelaties. Naast de standaard meegeleverde ondersteuning voor tal van data typen. Protocollen, document beveiligingen en ook partner management tools zijn beschikbaar.

Connectors

De connectoren van Crossworlds zijn meegeleverde interfaces naar andere applicaties en bieden diensten zoals transport, transformatie en event detectie. Crossworlds connectoren verzorgen de volgende zaken:

Verbinden van Applicaties: de implementatie van de interactie architectuur. Initialisatie, beëindiging, versie beheer en API linking.

Event notificatie: hiermee wordt bedoeld dat events kunnen worden opgevangen door de connectoren. Het type van het event en de nodige data worden herkend en opgevraagd bij de desbetreffende applicatie

Applicatie objecten en definities: hiermee is het mogelijk om op object niveau connectoren te ontwerpen en specificeren.

Connector development kit: met deze kunnen connectoren voor legacy of niet meegeleverde applicaties ontwikkelt worden

In de volgende lijst staan de meegeleverde connectoren ingedeeld in categorie software:

Enterprise Resource Planning (ERP)

Baan

JDE

Oracle

PeopleSoft

QAD

SAP

Customer Resource Management (CRM)

Clarify

Remedy

Siebel

Trilogy

Vantive/PeopleSoft

eBusiness

Ariba

BroadVision

CommerceOne

IBM Websphere

SAP BBP

Supply Chain Management (SCM)

i2

IMI

Manugistics
Numetrix/JDE
Technology
Corba
E-mail
HTTP
JDBC
MQSeries Integrator
ODBC
Oracle RDBMS
Text
Telecommunications
Kenan
Metasolv
Portal

Business objects

Dit is het interne dataformaat van Crossworlds. Voor de meeste algemene bedrijfsprocessen zijn er opnieuw te gebruiken *business objects* beschikbaar. Men moet het zien als een abstractie laag tussen procesgerelateerde en bedrijfsbrede veranderingen en applicatie specifieke veranderingen.

Collaboration en connectoren interacteren door het verzenden en het ontvangen van business objects. Een business object kan optreden als event, aanvraag (*request*) of antwoord (*respons*).

Event.

Een business object kan een event rapporteren. Als een connector een event ontdekt en een business object naar een collaboration stuurt speelt het business object de rol van het event.

Request.

Request komen in twee soorten voor:

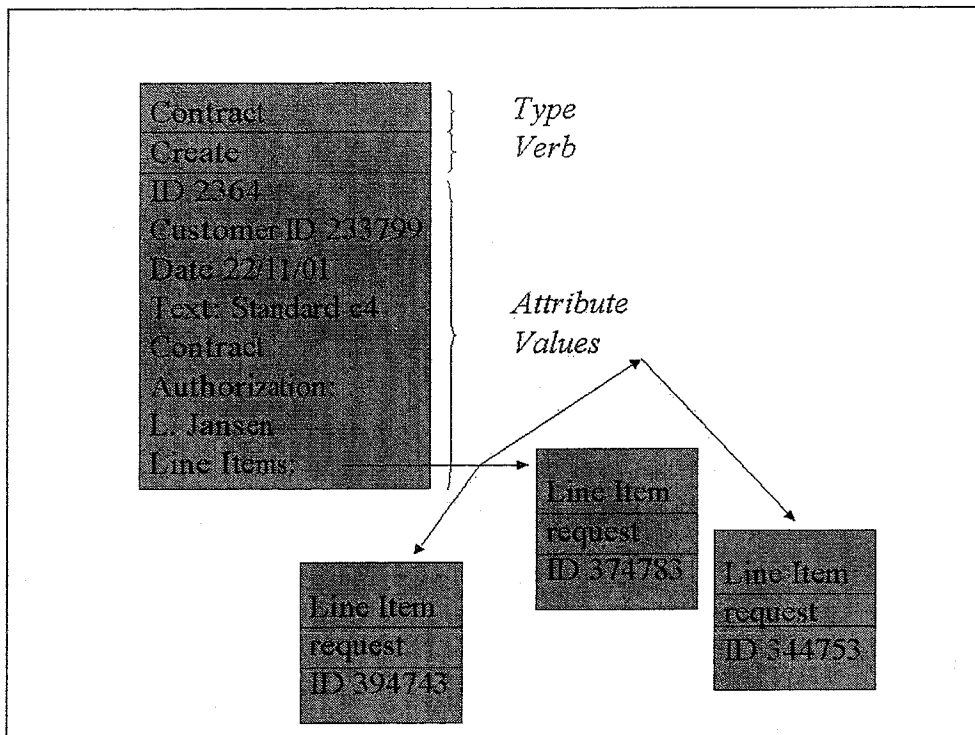
- Een collaboration kan een business object sturen naar een connector met daarin de opdracht om data in applicatie te plaatsen, te veranderen, te verwijderen of te verkrijgen.
- Het Server acces interface, deze wordt besproken in de volgende paragraaf, stuurt een business object als request naar een collaboration; als deze gemaakt is om te reageren wordt deze getriggered.

Response

Als een connector een request afgehandeld heeft stuurt het een respons.

Een business object bevat een type, de naam van het object, instructies voor het verwerken van het object, *verb* in de Crossworlds technologie en data en attributen. Deze attributen kunnen zelf ook weer *business objects* zijn.

Hieronder is een voorbeeld van een business object:



Server Access interface

Deze interface wordt gebruikt om aanvragen van externe systemen, zoals web-servers, te verwerken. Het is een API volgens de CORBA norm. dat synchrone data van zo wel interne als externe systemen ontvangt. De data wordt dan in *business objects* geconverteerd die weer door collaborations gebruikt kunnen worden. Nu de benaderingsmethoden van de server beschreven zijn kunnen we concluderen dat op volgende wijzen data opgeroepen kan worden:

Typen data calls

Er wordt gebruik gemaakt van vier verschillende communicatie vormen:

1. Asynchroon tussen applicaties via het netwerk
2. Calls van een web server naar een applicatie op het netwerk
3. Asynchrone request van applicaties via het lokale net naar externe webservers
4. Een *Collaboration* wat een sequentie van een of meerdere van de vorigen is.

De interacties die hiervoor nodig zijn worden gerealiseerd met drie modellen:

- Publish en subscribe

Rare zinnen; wat bedoel je nou? Een proces subscribed voor een event dat zijn uitvoering kan triggeren, dat doet dat proces door dat event te requesten en af te wachten. Bijvoorbeeld subscribed het proces voor het event *Employee.add()* en wacht tot de uitvoering van het proces tot dat het event aankomt. Een connector kan het proces triggeren door het event te publishen, het beschikbaar te maken. Er kunnen meerdere processen getriggered worden.

- Call

Een proces kan reageren op directe calls via het interface. Deze calls kunnen dus uit externe bronnen komen. Dit model wordt gebruikt als synchrone communicatie belangrijk is: bijvoorbeeld bij het opvragen van een voorraad niveau door een klant.

- Request en respons.

Een getriggered proces kan een request naar connectoren sturen en respons ontvangen.

Dit kan zowel met connectoren die communiceren met het netwerk, als ook met connectoren voor speciale data-typen, zoals bv XML die dan weer met de buiten wereld in verbinding staan.

Voorbeeld

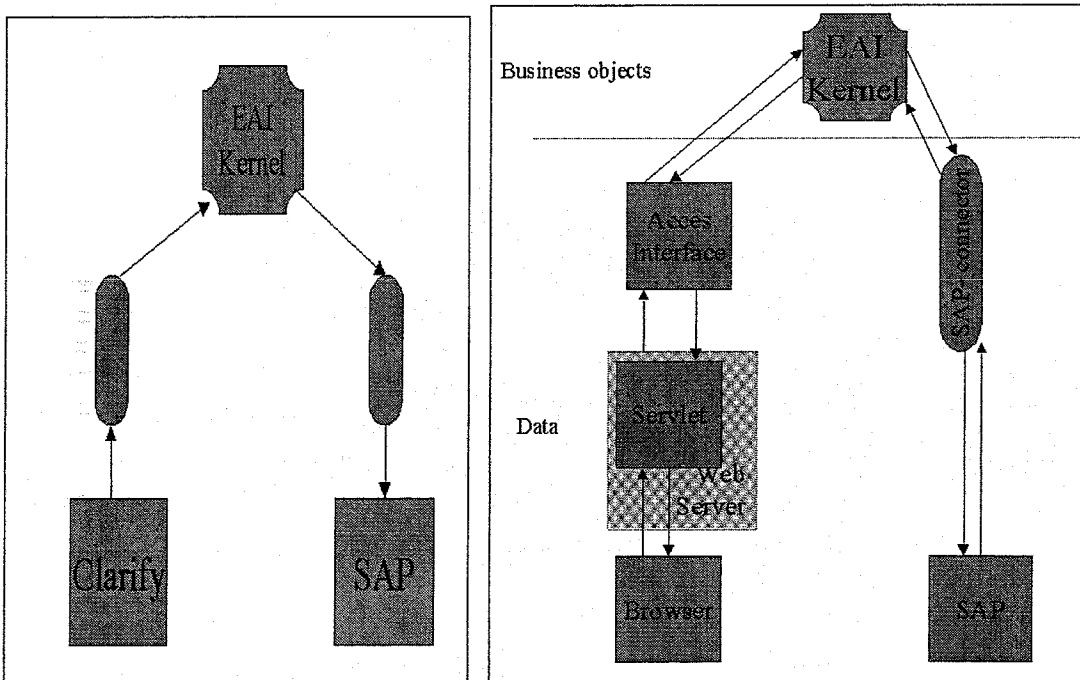
In dit voorbeeld wordt gebruik gemaakt van de hier boven genoemde componenten.

In de kernel uitleg, waar komt deze term nu ineens vandaan? bevindt zich een rules-base: hier definieert men hoe bepaalde gedistribueerde bedrijfsprocessen afgehandeld worden in collaborations. Externe calls kunnen bijvoorbeeld via een browser opgevangen worden door een acces interface afgehandeld. Andere verbindingen worden verwerkt door connectoren.

In de transformatie- en conversiecomponent worden dataformaten opgeslagen en vertaald.

In deze oplossing moeten zowel de kernel als de transformatie en conversiecomponent programmeerbaar zijn. Het is mogelijk om nieuwe data typen verwerken maar deze moeten kunnen worden gedefinieerd om ze te kunnen converteren in de interne taal van de applicatie: de business objects.

De kernel zou dan als volgt moeten werken: Als voorbeeld nemen we het veranderen van klant gegevens in de clarify applicatie, die zou moeten resulteren in een update in het SAP systeem. Er moet dus in de kernel een proces zijn : 'SynchroniseCustommers', die dit afhandelt. De twee connectoren voeren de vertaling uit van Clarify data naar business object en van business object naar SAP data.

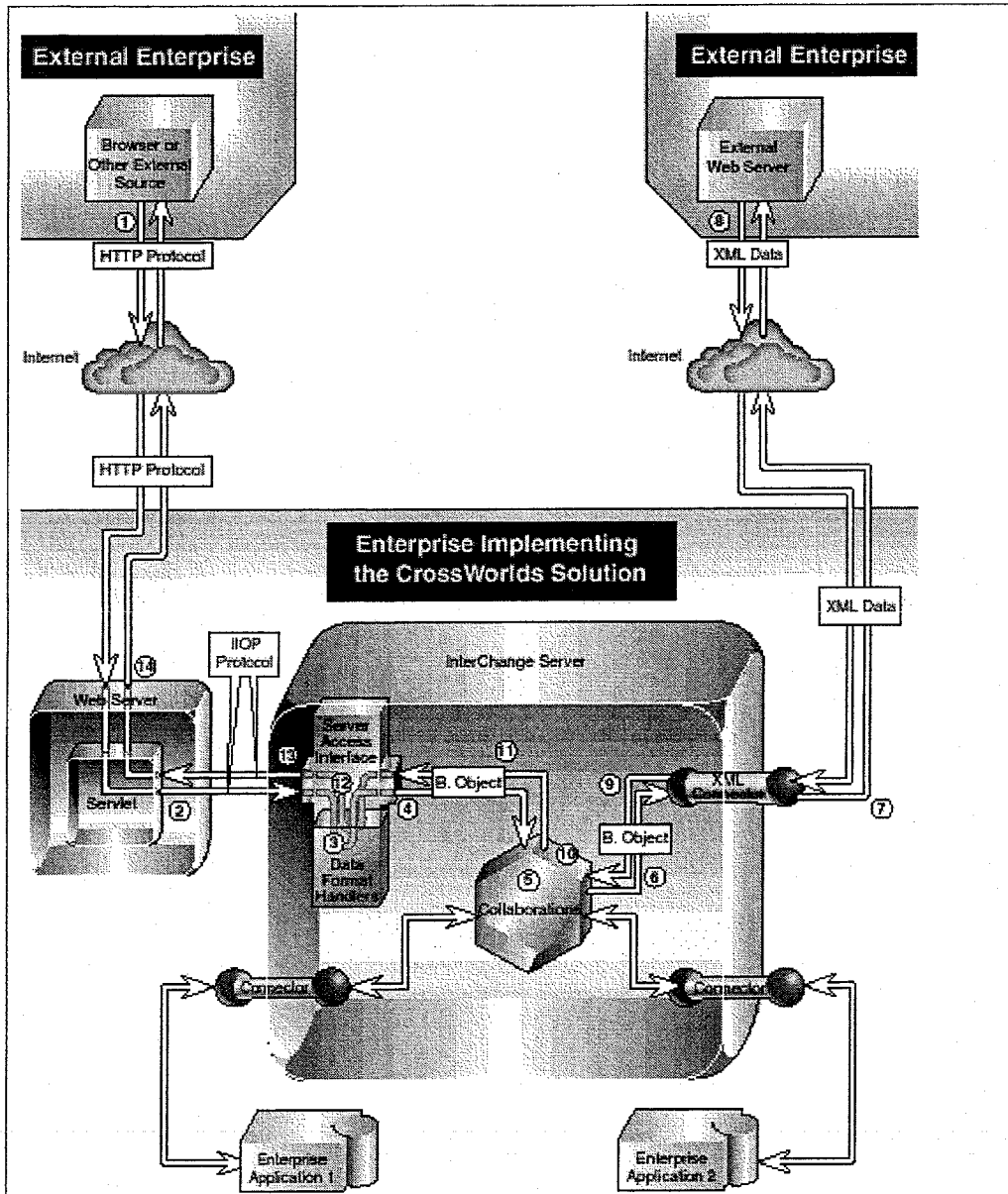


Een high level data flow voorbeeld:

Om inzicht te geven in hoe een dergelijke applicatie werkt wordt hieronder een schets gegeven van het aanvragen van XML document door een externe partij. Dit document bevindt zich niet op de locatie waar de Crossworlds Applicatie draait:

1. Een externe web browser plaats een call via het https protocol (bv een xml document)
2. Deze wordt door een servlet op de interne webserver opgevangen en afgehandeld, het servlet specificeert een business object en creëert het, de call triggert een actie bij de kernel. Het servlet gebruikt het IIOP protocol om de call door te sturen naar het interface van de server.
3. Het interface vangt de call op en laat de data vertalen door de conversie en transformatie component, in dit geval van XML naar een business object, en geeft het terug aan het interface
4. Het interface stuurt het business object naar de kernel
5. De kernel voert een bedrijfsproces uit op het business object en stuurt een exceptie of het resultaat terug naar het interface.
6. In dit voorbeeld stuurt de kernel een business object als request naar de XML-connector
7. De connector vertaalt het object in een XML document en stuurt die naar een web-server
8. De web server stuurt een respons naar de XML connector

9. De XML-connector vertaalt deze weer in een business object en stuurt het naar de kernel
10. De kernel voert het bij het business object horende proces uit
11. De kernel stuurt een business object naar het interface
12. Het interface laat de transformatie component het object vertalen en ontvangt het terug
13. Via het IIOP protocol stuurt het interface een bericht naar het servlet
14. Via het https protocol wordt de data naar de web-browser gestuurd



Het model

De hierboven beschreven diensten en componenten passen naar aanleiding van de gegevens op volgende manier in het model:

<p>Inter-Enterprise Process Automation Trading Partner Interchange. Ondersteund XML, EDI, http, https Partner management tools</p>	<p style="writing-mode: vertical-rl; transform: rotate(180deg);"> Application Interaction Server Access Interface en Interchange Server Publish/subscribe, request/response, call, ODBC, JDBC. </p>	<p style="writing-mode: vertical-rl; transform: rotate(180deg);"> Adaptor's Connectors Modular Non-proprietary Run-time extensibility van applicaties en DB mogelijkheid om zelf onderzocht te ontwikkelen </p>
<p>Business Process Automation Collaboration en Process designer. Met de designer kunnen collaborations een zoning grafisch ontworpen worden. De processen kunnen parallel uitgevoerd worden vanwege de multithreaded omgeving</p>		
<p>Integration Logic Collaboration en Interchange server. Content based routing. Helpde stroom van de berichten. Event sequencing. Multiple servers. Multithreaded. Hoge performance</p>		
<p>Application Content Transformation Server access interface en connectors. Visuele mapping tools. Voor gedefin- ieerde mappings. Direct many to many</p>		
<p>Communication Services Men maakt gebruik van Mercator TCI en IBM MQseries</p>		
<p>Infrastructure foundation NI en verschillende Unix versies. Java based InterChange Server TCP/IP en IIOP. Hub & Spoke</p>		
<p>MetaData Services Business Objects. Worden gemanaged in een <i>repository</i>. Object georiënteerde constructie</p>		
<p>Operational Services Er zijn Tools beschikbaar voor Analyse, monitoring, rapportage en load balancing. Men gebruikt Veritas als monitor. Beveiliging via SSL en herstart bij fout. Back ups via de <i>repository</i></p>		

