

TRIAD Arbetsrapport K 17

Krav på IA
Modellering
Utbildning för handledare
Katalogprinciper
Uttagssystem
Informationspridning

Rapport K nr 1: IRDS
Rapport K nr 2: IRDS Modeller och modellnivåer
Rapport K nr 3: Koppning begreppsmodell - relationsmodell
Rapport K nr 4: IBM:s Repository Manager- en Introduktion
Rapport K nr 5: IBM:s Repository Manager: Datamodelleringsbegreppen
Rapport K nr 6: IBM:s Repository Manager: Begreppsmodellering i Information Model
Rapport K nr 7: IBM Repository Manager: Attribut- och värdmodellering i Enterprise Submodel
Rapport K nr 8: Navigering i Repository
Rapport K nr 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget
Rapport K nr 10: TRIAD Newsletter –ISO/IRDS. Händelseutvecklingen 91/92
Rapport K nr 11: Samverkan mellan resurskataloger – visioner eller behov
Rapport K nr 12: AD/Cycle I Information Model – Processer och informationsflöden mellan processer
Rapport K nr 13: AD/Cycle I Information Model – Info Flows inom Processmodellen
Rapport K nr 14: AD/Cycle I Information Model – Relationsdatabasmodellering
Rapport K nr 15: AD/Cycle I Information Model – Härledningsspecifikationer i begreppsmodellen
Rapport K nr 16: IA-prototyp
Rapport K nr 17: Repository AD/Cycle IUG 1991

Repository AD/Cycle – IUG 1991

Björn Norén, Televerket Nätdivisionen

Spridningsförbehåll:

Denna rapport får endast spridas och användas inom de organisationer som deltar som parter i TRIAD-projektet.
© TRIAD-parterna okt 1992.

Rapporten är skriven i och för TRIAD delprojekt Katalogprinciper.

TRIAD-projektet drivs gemensamt av Televerket, Posten, Statskontoret, Ericsson Data Services och SISU, Svenska Institutet för systemutveckling.

För vidare information kontakta SISU Informationscentrum, tel 08-752 16 00, fax 08-752 68 00, Box 1250, 164 28 Kista.

Repository AD/Cycle (R/AD) International Users Group

Innehåll

Allmänt	2
Repository AD/Cycle for Enterprise Systems.	3
A Framework for Information Systems Architecture	4
Storing DB2 Definitions in the IBM Repository.....	6
Paradigm Modelling: Setting the Context for Everything.	7
Repository Manager: A Primer.	8
Moving Structured Projects to Object-Orientation	8
Enabling AD/Cycle.	9
Total Quality Management: The Context for AD/Cycle.	10
Model Versioning.	10
How to avoid a CASE disaster.	10
Objekt-Oriented Information Engineering -Food for Thought.	11
Application Development Productivity at Du Pont.....	12
Management of Knowledgeware's Mainframe Encyclopedia.	13

Repository AD/Cycle (R/AD) International Users Group

Second Annual Conference, Oct 27-30, 1991

Allmänt

Konferensen innehåll fem parallella presentationer och fem gemensamma och pågick under tre dagar. I anslutning till konferensen fanns en produktutställning. Det var dock dåligt med Repository produkter. Endast MSP och Reltech var representerade.

Ingen av presentationerna behandlade praktiska erfarenheterna med IBM's Repository. Bland deltagarna träffade jag i alla fall en användare eller före detta användare. Han hade lämnat tillbaka produkten. Skälet till detta var hög kostnad, inga CASE-verktyg kunde ännu anslutas och att man inte hade satt till resurser inom företaget för utvärdering.

IBM's Repository är ju inte avsett att användas "stand alone" utan i kombination med externa verktyg. Tills verktyg som fungerar tillsammans med Repositoryt finns framme kan man nog säga att det inte går att använda. Hur går det om CASE-leverantörerna utvecklar egna repositories? Kan man räkna med att IBM's informationsmodell kommer att bli "de-facto" standard eller kommer CASE-leverantörerna att göra för stora individuella avsteg från den att det inte blir någon standard? Det får tiden utvisa.

Den enda presentationen av praktisk användning av ett Repository var från Allstate Insurance Company (se referat nedan). Repositoryt var ADW's och föredraget gav intryck av att det vara ett mycket tungt arbete med införa ett Repository och många problem återstår att lösa.

Tungdpunkten på föredragen låg därför på metodfrågor. Föredragens innehåll och antal (grovt uppskattat):

IBMs Repository	6
Metod, erfarenheter mm	16
CASE	5
Objektorientering	5
Vision, arkitektur	3
Övrigt	2
<hr/>	
Summa	37

Nedan följer ett sammandrag av några av föredragen:

Repository AD/Cycle for Enterprise Systems.

Dr. Robert H. Holland

Holland behandlade systemutvecklingen ur ett affärsperspektiv.

Organisationer blir nu mera marknadsorienterade och utsatta för kortare produktlivscyklar, smalare marknadssegment, global konkurrens, förändrade marknader, statlig avreglering, förändrade organisationsformer, etc.

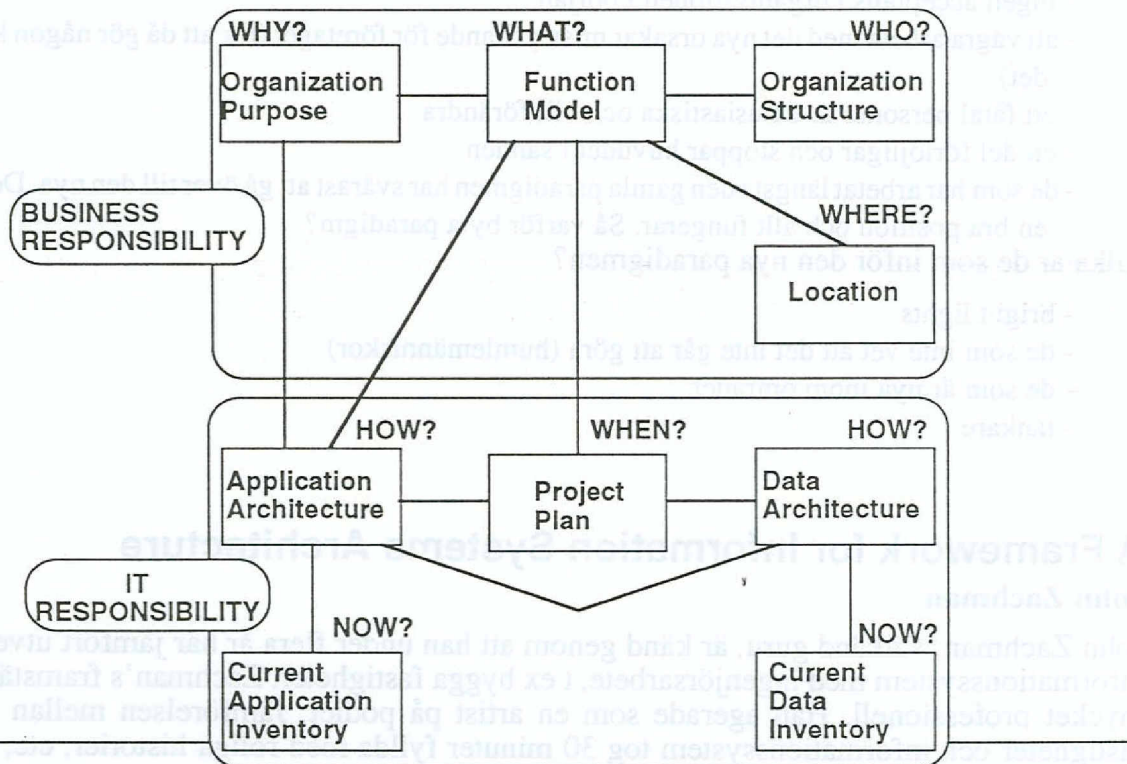
De försöker därför använda samma ramverk (infrastruktur) för olika typer av produkter. T ex Toyota använder sitt ramverk från sin biltillverkning till att börja tillverka flygplan.

IT ska därför inte längre betraktas som ett "cost-center" utan som ett "investment-center". IT ska inte bara användas för att reducera kostnader utan också för att skapa intäkter genom "value-added services".

Företagsledningen allokerar resurser till IT och har det övergripande ansvaret för IT. IT tillhandahåller ramverket och tekniken och visar på möjligheterna.

Holland visade ett flertal bilder på detta tema. En strategisk bild över CASE visas som exempel nedan:

CASE Strategy Data Structure



Systemutvecklarna kan inte längre säga att de inte har tid att göra rätt från början, men att de alltid har tid att fixa till det senare.

AD/Cycle och Repository Manager innebär ett paradigmskifte. Holland tog upp vissa aspekter på att lyckas med detta. Att skifta paradigmen innebär inte något kärriärlyft för datachefen, så det ligger ofta inte i hans intresse.

För att lyckas behövs bl annat "Bright lights". "Bright lights" ser längre än andra. Övriga följer efter och anpassar sig eller ser inget ljus alls utan talar bara om gamla tider.

För att kunna planera måste man först lysa upp framtiden.

För att lyckas måste en organisationen kunna:

- förutse
- införa nyheter
- vara främst i kvalitet

Ett "bright light" måste kunna:

- påverka förståelse när förståelse påverkar
- tänka i okonventionella banor
- jämka samman och prioritera på önskat resultat
- upprätta en migreringsväg
- visa på en metod för problemlösning

Paradigmskiften sker stegvis. De börjar med att "bright lights" ser mönstret. Sen ser övriga organisationen också mönstret och tillsätter folk för att lösa problemen.

Det återstående problemet med det praktiska införandet är det svåraste att lösa. Organisationen frestas att överlåta detta till en extern organisation. Holland varnade för det, det är att fly undan.

Följande observationer har gjorts vid paradigmskiften:

- ingen acceptans i organisationen i början
- att vägra arbeta med det nya orsakar misslyckande för företaget (för att då gör någon konkurrent det)
- ett fåtal personer är entusiastiska och vill förändra
- en del förlöjligar och stoppar huvudet i sanden
- de som har arbetat längst i den gamla paradigmen har svårast att gå över till den nya. De har redan en bra position och allt fungerar. Så varför byta paradigmen?

Vilka är de som inför den nya paradigmen?

- bright lights
- de som inte vet att det inte går att göra (humlemänniskor)
- de som är nya inom området
- tänkare

A Framework for Information Systems Architecture

John Zachman





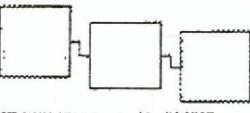
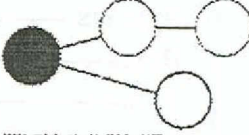
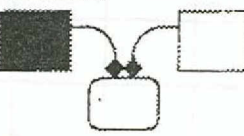
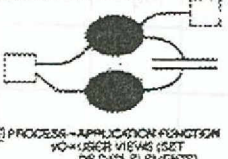
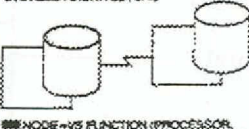
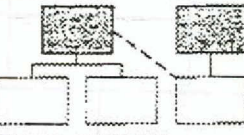
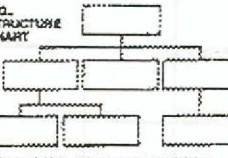
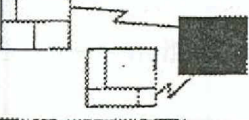



John Zachman, välkänd guru, är känd genom att han under flera år har jämfört utveckling av informationssystem med ingenjörsarbete, t ex bygga fastigheter. Zachman's framställning var mycket professionell. Han agerade som en artist på podiet. Jämförelsen mellan att bygga fastigheter och informationssystem tog 30 minuter fyllda med roliga historier, etc, men med mycket lite detaljer.

Zachmans ramverk finns beskrivet i IBM. System Journal, vol 26, no 3, 1987, och går ut på att man kan se utvecklingen av informationssystem utifrån de inblandade parternas olika perspektiv. Den översta nivån ses ur kundens perspektiv. Under denna kommer systemerarens och konstruktörens perspektiv. Kunden vill ha en produkt som tjänar sitt ändamål. Kundens beskrivning översätts sedan till en beskrivning till systemerarens perspektiv och så vidare.

De olika representationerna är inte en ökande nivå av detaljering utan en skillnad i innehåll, semantik, etc.

Det finns också olika typer av beskrivningar orienterade mot olika aspekter av vad som beskrivs. De grundläggande är data (vad?), funktion (hur?) och placering (var?). De två första är de kända data- och funktionsbenen. Zachman's "Extended" innehåller även personer (vilka?), tid (när?) och motivering (varför?).

Ramverket bildar en matris innehållande celler bestående av olika beskrivningstyper (metoder).

	DATA DESCRIPTION ■ ENTITY □ RELATION	PROCESS DESCRIPTION ■ PROCESS □ INPUT/OUTPUT	NETWORK DESCRIPTION ■ NODE □ LINE
SCOPE DESCRIPTION (BALLPARK VIEW)	LIST OF ENTITIES REPERTOIRE TO THE BUSINESS  ■ ENTITY = CLASS OF BUSINESS ENTITY	LIST OF PROCESSES THE BUSINESS PERFORMS  □ PROCESS = CLASS OF BUSINESS PROCESS	LIST OF LOCATIONS IN WHICH THE BUSINESS OPERATES  ■ NODE = BUSINESS LOCATION
MODEL OF THE BUSINESS (OWNER'S VIEW)	E.G. ENTITY RELATIONSHIP DIAGRAM  ■ ENTITY = BUSINESS ENTITY □ RELN = BUSINESS RELN	E.G. FUNCTIONAL FLOW DIAGRAM  □ PROCESS = BUSINESS PROCESS ■ IO = BUSINESS RESOURCES	E.G. LOGISTIC NETWORK  ■ NODE = BUSINESS UNIT □ LINE = BUSINESS RELATIONSHIP FLOW
MODEL OF THE INFORMATION SYSTEM (DESIGNER'S VIEW)	E.G. DATA MODEL  ■ ENTITY = DATA ENTITY □ RELN = DATA RELATIONSHIP	E.G. DATA FLOW DIAGRAM  □ PROCESS = APPLICATION FUNCTION ■ IO = USER VIEWS (SET OF DATA ELEMENTS)	E.G. DISTRIBUTED SYSTEMS ARCHITECTURE  ■ NODE = FS FUNCTION (PROCESSOR, STORAGE, ACCESS, ETC.) □ LINE = LINE CHARACTERISTICS
TECHNOLOGY MODEL (BUILDER'S VIEW)	E.G. DATA DESIGN  ■ ENTITY = SYSTEM FLOW □ RELN = POWER RELN	E.G. STRUCTURE CHART  □ PROCESS = COMPUTER FUNCTION ■ IO = SCREEN/DEVICE FORMATS	E.G. SYSTEM ARCHITECTURE  ■ NODE = HARDWARE/SYSTEM SOFTWARE □ LINE = LINE SPECIFICATIONS
DETAILED DESCRIPTION (OUT-OF-CONTEXT VIEW)	E.G. DATA BASE DESCRIPTION  □ ENTITY = FIELDS ■ RELN = ADDRESSES	E.G. PROGRAM  □ PROCESS = LANGUAGE STATEMENTS ■ IO = CONTROL BLOCKS	E.G. NETWORK ARCHITECTURE  ■ NODE = ADDRESSES □ LINE = PROTOCOLS
ACTUAL SYSTEM	DATA	FUNCTION	COMMUNICATIONS

För framtiden ansåg Zachman att systemutvecklare var tvungna att tänka om. De måste lära sig att tillverka standardkomponenter och med hjälp av dessa snabbt kunna tillverka (plocka ihop) mot order i likhet med övrig fabriksstillverkning.

Det är också målet för AD/Cycle men det kräver ett paradigmskifte.

Storing DB2 Definitions in the IBM Repository.

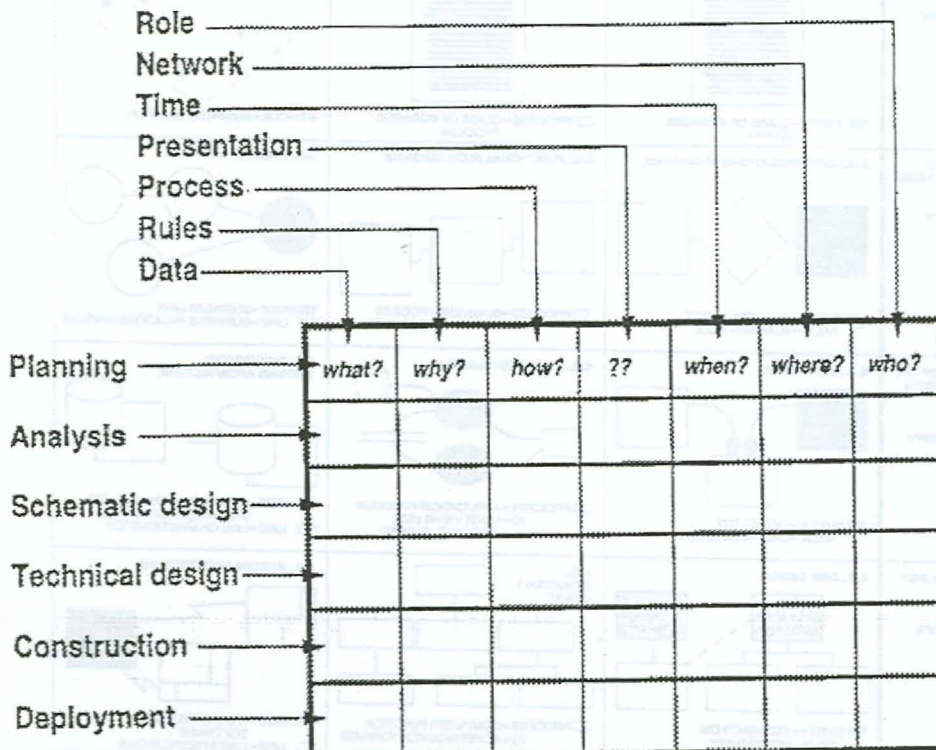
Chris Loosley

Loosley har, tillsammans med Zachman, utvecklat Zachman's arkitektur till "Zachman Extended".

I den ursprungliga arkitekturen fanns kolumnerna Data (what?), Funktion eller process (how?) och Nätverk (where?).

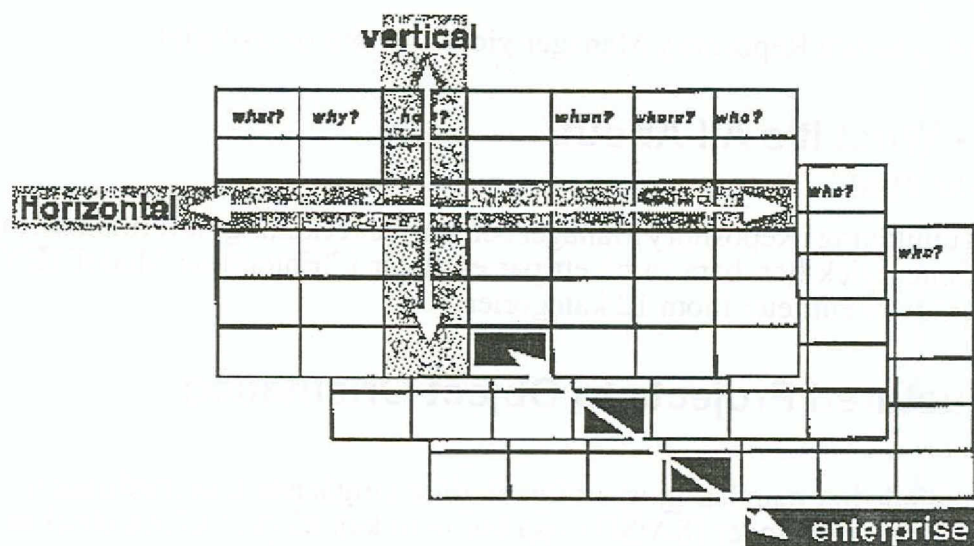
De har nu lagt till Role (who?), Time (when?), Presentation och Rules (why?).

The DA/Ze Framework (DataBase Associates / Zachman extended)



Över arkitekturen kan man lägga olika aspekter på integrering:

- Horizontal, t ex integrering mellan funktion och data
- Vertical, t ex integrering över livscykeln
- Enterprise, t ex integrering och återanvändning av konceptuella datamodeller för olika system
- Control, t ex likartad funktionalitet i verktyg
- Visual, t ex likartad presentation



Loosley redovisade också en jämförelse mellan IBM's Common Relational Database (DRC) Submodel och DB2's katalog.

Paradigm Modelling: Setting the Context for Everything.

Matthias W.N. Benfey, Royal Bank of Canada.

Det har talats mycket om "paradigmskiften". Föredraget behandlade en modell och en metod med vars hjälp paradigmtänkande kan appliceras för att avgöra beskaffenheten och sammanhanget av olika frågor.

Talet om paradigmskiften har blivit aktuellt på grund av de massiva förändringar från en konfigurering av förhållanden till en ny som nu sker i samhället. Gamla tankesätt gäller inte längre. Den lyckas som vet vad som ska förändras och när det är dags.

Ett exempel på en paradigm är Zachman's arkitektur. Benfey kopplar den till omvärlden i tre plan, den globala ekonomin, företagets uppgift och målmarknader. Detta kopplas till arkitektur-ens olika celler. Det är denna koppling som är modellen för "paradigm modellering".

Tre "kopplingar" bildar kärnan i paradigm-modellering:

- sammanhang/innehåll
- likhet/olikhet
- kontinuitet/diskontinuitet

Benfey visade praktiska exempel på paradigm-modellering med exempel från Apple och Yamaha.

Repository Manager: A Primer.

Joe Breal, IBM

Grunderna i AD/Cycle och Repository Manager gicks igenom översiktligt.

Info Model - What It's All About.

Patti Cartwright, IBM

Patti arbetar med utbildning i Repository Manager och är mycket kunnig i informationsmodellen. Under presentationen gick hon bara in på ett par entiteter i "Enterprise Model"(ENT). Den består i dagsläget av 66 entiteter inom 12 kategorier.

Moving Structured Projects to Object-Orientation

IBM, Minnesota.

Föredraget behandlade hur man har gjort ett försök till att utgående från traditionella strukturerade analys- och designmetoder (SA/SD) visa hur man kan gå över till en objektorienterad implementering (design och programmering).

Då slipper man kasta ut allt traditionellt och börja om från början.

Med traditionella analysmetoder avses främst dataflödesdiagram (DFD) med processspecifikationer och ER-modellering.

Analysresultatet transformeras i två steg:

- Det första identifierar kandidatobjekt (tänkbara objekt) ur analysmodellen.
- Det andra är klassyntes och förfining.

Kandidatobjekt hittas genom "Carving". "Carving" består av att skära ut kandidatobjekt ur DFD-et på dess lägsta nivå. Kandidatobjekten namnges och dess operationer listas. Hur man praktiskt bär sig åt beskrivs för varje komponent i dataflödesdiagrammet: "External Entity", "Data Store", "Flow" och "Process".

Sedan övergår man till programspecifikationerna som är beskrivna i strukturerad engelska. I dessa ringar man in de tidigare funna kandidatobjekten.

Nästa steg är klassyntes. Syntesen består i att infoga kandidatobjekten i den befintliga klasshierarkin. Klassificering av objekt görs genom att jämföra instanserna för att se vilka som hör till samma typ. Instans används synonymt med objekt och typ synonymt med klass. Två objekt kan tillhöra samma klass om de har likvärdiga operationer.

I detta arbetssteg ingår också förfining vilket kan resultera i delning och ihopslagning av klasser.

Med hjälp av processspecifikationerna skapas kandidatmetoderna.

Implementeringen görs i ett objektorienterat programmeringsspråk t ex C++.

Man nämnde att man inte klart såg nyttan av ER-diagrammen vid objektorienterad programmering och det framgår inte heller hur ER-diagrammen behandlas. Skälet till detta kan vara att man inte talade om implementering i någon typ av databas. Sambandet mellan dataflödesdiagrammet och ER-diagrammet framgår inte heller i den traditionella analysmetoden.

Enabling AD/Cycle.

Graham Thompson, MSP

Föredraget beskrev olika frågor som en organisation ställs inför vid införandet av AD/Cycle.

Det inleddes med en beskrivning av AD/Cycle. AD/Cycle är ett koncept eller ett ramverk, inte en produkt eller en samling produkter. Konceptet är inte nytt, IBM har bara legitimerat det och pekat ut riktningen.

Ramverket består av tre nyckelområden för att åstadkomma en framgångsrik ADE (Applikation Development Environment) bestående av:

1. En Repository plattform

- för att styra och kontrollera all information inom systemutvecklingsprocessen.
- reflekterar organisationens metoder och verktyg.

2. En öppen verktygsarkitektur

- för val och integrering av olika verktyg.

3. Ett livscykelstöd

- för projektstyrning integrerad med ADE.

Man ska inte skapa bryggor mellan olika verktyg utan istället bryggor mellan verktyg och Repository. Bryggorna skapas genom att jämföra Repository's informationsmodell med verktygets och skapa en "interface"-informationsmodell. Denna utgör grund för eventuell realisering och uppdateras i samband med nya Repository- och verktygsversioner.

För att beskriva sin organisations nuläge inom AD/Cycle konceptet kan man ta ett tomt diagram över AD/Cycle arkitekturen och i detta skriva in sina verktyg. Ett förslag på en AD/Cycle implementeringsstrategi beskrevs i presentationen. En fullständig implementering ansåg man skulle ta minimum fem år!

Fördelar man uppnår är:

- automatisering av arbetstegen inom systemutvecklingsprocessen.
- produktivitetsvinster.
- återanvändning.
- kvalitetshöjning.

MSP ansåg att det kommer att ta tre till fem år innan IBM's Repository är moget att stödja en operationell ADE och att man därför under tiden ska välja ett interimistiskt Repository (MSP's?).

Följande fem nyckelområden bör undersökas vid en utvärdering:

- Repository's informationsmodell
- Import/export möjligheter
- Versions- och konfigurationshantering
- Audit och säkerhet
- Migreringsväg till Repository Manager

Införandet ska betraktas som ett betydande projekt och man ska i detta använda samma projektstyrnings- och systemutvecklingsmetoder som för ett vanligt stort systemutvecklingsprojekt.

Införandet kommer inte att ske över natten utan kommer att vara ett långt, hårt slit.

Total Quality Management: The Context for AD/Cycle.

Vaughan Merlyn, Ernst & Young.

Ernst & Young har i USA inget förhållande med Knowledgeware (som i Sverige).

Fyra nyckelområden för kvalitet:

1. Innovationer i produkter och tjänster
2. Innovationer i hanteringsprocessen
3. Förbättringar av existerande produkter och tjänster
4. Förbättringar i hanteringsprocessen

Det är viktigt att arbeta med hanteringsprocessen. Skilj på innovationer och förbättringar. Merlyn ansåg att Repository Manager hör till innovationer. Han påtalade också vikten av kultur och paradigmer. Det gäller att förändra företagets mentala modeller. En modell för att se på affärsförbättringar och en metod för hur man går till väga presenterades samt några tips t ex:

- mät processen, inte människor
- mät ej om du inte gör något åt resultatet av mätningen
- fråga kunden vad du ska mäta, inte gurus
- inför en process för att förbättra processen
- om du inte har en kvalitetsenhet, tillsätt en
- om du har en kvalitetsenhet, avskaffa den

Model Versioning.

William Irwin, Titan

Många hade mött upp på denna presentation av ett länge känt problem. Tyvärr fick vi inte någon lösning på problemet denna gång heller.

How to avoid a CASE disaster.

Joe Armitage, Titan

Armitage redovisade några CASE myter och diskuterade dem:

Det finns en perfekt datamodell, och vi behöver bara analysera på tills den är klar.

Vi behöver inte ändra vår organisationsstruktur.

Vi behöver ingen metodik för att:

- vi har redan en
- den är inbyggd i verktyget
- metodik?.....Vad är det?

Vi behöver inte utbilda folk för att:

- mjukvaran är så användarvänlig (GUI)
- det är bara en bunt diagram
- leverantören utbildar oss på en vecka

Vi behöver inte dyra konsulter

Vi kan anställa mindre kvalificerade personer.

De goda nyheterna är:

CASE projektstyrning är mycket likt traditionell projektstyrning.

Du kan fortfarande använda ditt sunda förnuft!

De dåliga nyheterna är:

CASE projektstyrning är mycket likt traditionell projektstyrning.

Du måste använda ditt sunda förnuft!

Objekt-Oriented Information Engineering -Food for Thought.

Ron Schultz, Berard Software Engineering

Schultz diskuterade styrkor och svagheter med Information Engineering och Objektorientering. Diskussionen var inte på något sett avsett att vara uttömmande utan endast ett underlag för vidare diskussioner.

På 60-talet styrde dataavdelningen dataprogram, på 70-talet datorsystem, på 80-talet information. På 90-talet försöker man styra datoriserade företag. De facto metodstandard för detta i dag är IE. IE's mål är att ge rätt information till rätt personer vid rätt tidpunkt.

Schultz gav en kortfattad förklaring av IE och gick sedan in för att (subjektivt) jämföra IE med OO.

Att få ihop IE med OO är för närvarande ett utforskat område. IE ignorerar helt OO. IE fokuserar på "information" och OO på "objekt".

OO fokuserar på återanvändning av kod. IE fokuserar på att ur beskrivningar generera kod automatiskt. IE ignorerar mjukvarukvalitet, konfigurationshantering och testning. IE's starka sida har varit på planering och analys av informationssystem. Att realisera dessa system har inte varit lika framgångsrikt.

IE's stöd för distribution av data (t e x distribuerade databaser) är mycket primitivt eller obefintligt. IE presenteras ofta som en top-down helhetslösning där man uppmanas att ta allt eller inget. Att bygga en datamodell över ett helt företag och konvertera existerande data till denna modell och sen skapa applikationer som arbetar mot detta nya data är ett enormt arbete. IE ger minimalt stöd för återanvändning av komponenter eller för en inkrementell systemutveckling och systemintegrering.

OO teknik å andra sidan ger betydande stöd för inkrementellt systembyggande och återanvändning av mjukvarukomponenter.

James Martin definierar information som "any formal, structured data that is required to support a business and can be stored or retrieved from a computer". Men personer i verksamheten definierar information som de skärmbilder, rapporter, memos, etc som de behöver i sitt arbete. Strukturering av data kräver tillgång till en arbetsprocess utöver datamodellen i IE. Strukturering kräver att data sätts samman tillsammans med ett ramverk av affärsregler och formatregler. För dessa finns ingen naturlig hemvist i IE.

IE fokuserar på entiteter och sambandet mellan dessa. OO fokuserar på objekt. En entitet har skämtsamt beskrivits av Bob Brown från The Database Design Group som "ett degenererat objekt som ännu ej lärt sig bete sig". Denna definition visar på att objekt inte är data.

ER-modellering, ett kraftfullt verktyg för dataanalys, hjälper inte till att beskriva affärsregler eller datas beteende och dess information.

Fortsättningen bestod i en del, som jag anser svaga, oklara argument för objektorientering.

Application Development Productivity at Du Pont.

Richard B. Stromberg

Du Pont har 7500 personer som arbetar inom olika informationssystemsfunktioner. Föredraget behandlade Du Pont's förändrings- och paradigmskiftes- historia från 60-talet till nutid.

Paradigm definieras som en mängd gränser och reglerna för att arbeta inom dessa gränser. Med paradigmskifte menas att gå tillbaka till noll och starta om inom en ny paradigm. För att klara av paradigmskiften måste man lära av historien för att inte göra om samma misstag igen.

Stromberg gick igenom de paradigmskiften Du Pont gått igenom sedan 60-talet med avseende på bland annat arkitektur, metodik och verktyg.

För 90-talet har Du Pont en modern systemutvecklingsmodell som är iterativ i likhet med Sveriges "SIS-modell". I den är datamodellering och prototyping viktiga nya beståndsdelar.

Arkitekturen består av en integrerad, top-down modelldriven ansats i likhet med Information Engineering. Standardmetoder finns för varje cell i Zachman's "Framework for Information Systems Architecture". Guidelines för verktyg finns för systemutvecklingens olika faser. För varje verktyg finns ansvarig enhet med ansvar för leverantörskontakter, distribution av nya verktygsutgåvor, installation, "hot line" och konsultstöd.

En central enhet för dataadministration med mer än 100 personer har inrättats. Den ansvarar för "Du Pont Reference Model", innehållande företagets verksamhetsfunktioner och begreppsområden, en matris mellan dessa, en konceptuell datamodell och processmodeller med hjälp av dataflödesdiagram.

Dataadministrationsfunktionen ansvarar även för "Reference Data Bases" (ämnesdatabaser), dataelementstandard, företagskoder, modelleringsstandard, datakatalog i Data Manager, verktygs-guidelines, verktygsstöd och konsultstöd inom dataadministration.

Management of Knowledgeware's Mainframe Encyclopedia.

Kathleen Syverson, Allstate Insurance Company.

Många som har skaffat en stordatorencyklopedia har upptäckt att det saknas en metod över hur man ska använda den.

Ett företag har dock utvecklat anvisningar för administration av en Encyklopedia. Dessa består av organisation av encyklopedian och dess stödpersonal, fysiskt underhåll, versionshantering, modellhantering och användarstöd. Vissa av anvisningarna är teoretiska, vilket innebär att de inte har provats praktiskt.

Dataadministrationsfunktionen ansvarar för all generell hantering av encyklopedian.

Encyklopedian är hierarkiskt organiserad i följande fyra nivåer: "Enterprise Architecture", "Business area", "Design area" och "Personal".

"Enterprise Architecture" nivå innehåller information från planerings- och analysfasen och är företags-gemensam information. Denna del underhålls av dataadministrationsfunktionen.

Nästa nivå, "Business area", innehåller alla objekt som behövs för att beskriva ett informationssystem. Detta är den högsta nivå där "Design area"-objekt lagras. För "Business area"-encyklopedian ansvarar den som är funktions-områdesansvarig för informationssystemet.

Den tredje nivån, "Design area"-encyklopedian, består av utdrag ur "Business area"-encyklopedian för etablering av ett projekt. Funktionsområdesansvarig ansvarar även för denna nivå.

Den fjärde nivån, "Personal"-encyklopedian, är PC-baserad och innehåller ett utdrag ur "Business

area"-encyklopedian eller "Design area"-encyklopedian beroende på typ av projekt. Projektet ansvarar för denna nivå.

För att hålla reda på de olika encyklopedierna används ett standardiserat filnamngivningssystem innehållande ett två siffrors versionsnummer för manuell versionshantering. Versionshantering finns inte i ADW/MVS.

Ansvar för underhållet fördelas på de olika nivåerna ovan, med ansvariga dataadministratörer på företags-, "Business area", "Business area unit" och projekt.

Utbildningen har bestått av en 13 dagars metod- och verktygskurs.

ADW/MVS saknar funktioner för filunderhåll och detta har orsakat mycket extraarbete.

Ändrings- och versionshantering kräver fasta rutiner och innebär mycket arbete med detaljer för att upprätthålla dataintegritet. Den praktiska erfarenheten av detta är låg. Några projekt har valt att inte uppdatera stordatorencyklopedian p.g.a. av att det innebär för mycket merarbete med filöverföring fram och tillbaka mellan PC:n och stordatorn.

Modellhanterings främsta uppgift att lösa objektkonflikter mellan olika modeller. Det är en intellektuell uppgift och ingen verktygsfunktion men verktyget ska ge stöd. ADW/MVS's sätt att organisera encyklopedian i separata delencyklopedier har inte givit det stöd som man hade trott från början. Man behöver desperat mapping mellan de olika delencyklopedierna.

Man behöver funktioner för bättre ändringskontroll, objektförändringshistoria, encyklopediaförändringsanalys, encyklopedia jämförelsefunktioner och bättre rapportmöjligheter.

TRIAD-rapporter per 921130

Verksamhetskrav på informationsadministration

- V 1: IA och verksamhetens krav – erfarenheter från offentlig förvaltning
- V 2: Fallstudie av IA-projektet vid Televerket
- V 3: IA-erfarenheter från företag och myndigheter
- V 4: Frihandel och handlingsfrihet - informationsvägar till 2000-talet (**KOMMER jan 1993**)

Modellering

- N 1: Modelleringsansatser för begrepps- och datamodellering: – Beskrivning och försök till jämförelse
- N 2: Generering av konceptuella modeller från policydokument
- N 3: Espritprojektet Tempora
- N 4: Prövning av regelbaserad metodik inom Posten
- N 5: En kokbok i remodellering - utkast
- N 6: Datorstöd för modellintegration
- N 7: Modellbaserad kunskapsinsamling
- N 8: Modellkvalitet (**NY - nov 1992**)
- N 9: Samband mellan dokument och modeller (**NY - nov 1992**)

Utbildning

- H1 – Handledarutbildning för modelleringsledare, avancerad
- H2 - Slutrapport HUMLA prototyp (**NY - nov 1992. 4 sidor**)

Katalogprinciper

- K 1: IRDS
- K 2: IRDS Modeller och modellnivåer
- K 3: Kopping begreppsmodell - relationsmodell
- K 4: IBM:s Repository Manager- en Introduktion
- K 5: IBM:s Repository Manager: Datamodelleringsbegreppen
- K 6: IBM:s Repository Manager: Begreppsmodellering i Information Model
- K 7: IBM Repository Manager: Attribut- och värdemodellering i Enterprise Submodel
- K 8: Navigering i Repository
- K 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget
- K 10: TRIAD Newsletter –ISO/IRDS. Händelseutvecklingen 91/92
- K 11: Samverkan mellan resurskataloger – visioner eller behov
- K 12: AD/Cycle I Information Model – Processer och informationsflöden mellan processer
- K 13: AD/Cycle I Information Model – Info Flows inom Processmodellen
- K 14: AD/Cycle I Information Model – Relationsdatabasmodellering
- K 15: AD/Cycle I Information Model – Härlednings-specifikationer i begreppsmodellen
- K 16: IA-prototyp (**NY - nov 1992**)
- K 17: Repository AD/Cycle (R/AD) - International Users Group (**NY - nov 1992**)

Uttagssystem

- U1: Hybris i Unix-miljö, förstudie
- U2: DEBRIS (**NY - nov 1992**)